A MULTI-PHYSICS APPROACH TO THE CO-DESIGN OF 3D MULTI-CORE PROCESSORS

A Dissertation
Presented to
The Academic Faculty

By

He Xiao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2018

A MULTI-PHYSICS APPROACH TO THE CO-DESIGN OF 3D MULTI-CORE PROCESSORS

Approved by:

Dr. Sudhakar Yalamanchili, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Yogendra Joshi
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Saibal Mukhopadhyay
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Hyesoon Kim
School of Computer Science
Georgia Institute of Technology

Dr. Yorai Wardi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: January 11, 2018

To my wife Xueyi, and my parents for their endless love and support

ACKNOWLEDGEMENTS

This dissertation will not be possible without the inspiration, effort, and support from many people. First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Sudhakar Yalamanchili, for his endless generosity of guidance, support and patience throughout my Ph.D. studies. His vision, insight, and encouragement have greatly inspired my research, study, and life. I will forever be grateful to him. I also want to extend my thanks to Dr. Saibal Mukhopadhyay, Dr. Yogendra Joshi, Dr. Yorai Wardi, and Dr. Hyesoon Kim for serving on my committee and collaborating with a series of research and publications. It has been a great privilege to work with them. I would like to thank the former and current members in the CASL lab for their inspiration, mentoring, and collaboration. I am very lucky to be working with Dr. William Song, Dr. Jun Wang, Dr. Indrani Paul, Dr. Minhaj Hassan, Chad Kersey, and Xinwei Chen, and I appreciate the time together with Dr. Eric Anger, Dr. Jin Wang, Si Li, Karthik Rao, Dr. Jeffrey Young, Dr. Andrew Kerr, Si Li, and Karthik Rao. I wish to thank Dr. Zhimin Wan, Dr. Wen Yueh, Dr. Monodeep Kar, and Dr. Lifeng Nai for the collaboration and dedication. Last but not least, I would like to thank Dr. Arora Manish and Dr. Wei Huang from AMD Research for the internship opportunity and industry experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

BEOL back-end-of-line.

DVFS dynamic voltage and frequency scaling.

eDRAM embedded dynamic random-access memory.

EPI energy per instruction.

FinFET fin field-effect transistor.

HBM high bandwidth memory.

HTC heat transfer coefficient.

IC integrated circuit.

IPC instructions per cycle.

IVR integrated voltage regulator.

LLC last-level cache.

MOSFET metal-oxide-semiconductor field-effect transistor.

PDN power delivery network.

SRAM static random-access memory.

TDP thermal design power.

TSV through-silicon via.

VLSI very-large-scale integration.

SUMMARY

The three-dimensional (3D) integrated circuit becomes a promising solution to address the dark silicon effect for future processors. However, design of such 3D systems exposes great challenges to computer scientists. The objective of this thesis is to characterize the relationship between processor performance, thermal cooling, and power delivery in 3D systems and to optimize 3D many-core processors so as to extend Amdahl's Law for performance and power benefits. To understand the low-level circuital and physical behaviors in 3D integrated circuits, we construct detailed multi-physical models for the essential components of processors including the power regulator and cache arrays. These models, validated by commercial simulation software, are integrated into a cycle-based full-system simulation framework for large-scale many-core processors.

With this framework, we explore the co-design opportunities in 3D processors and demonstrate the necessity and feasibility of holistic optimization in 3D processors to achieve both performance gain and energy efficiency. Specifically, this dissertation consists of three research practices in 3D processors: i) a thermal-architecture co-design that improves cooling capability in 3D packages and introduces adaptation mechanisms in core logic and cache structure for performance and power efficiency based on our characterization on thermally dependent architectures under various cooling techniques; ii) a power-architecture co-design that minimizes voltage guard band and reduces runtime power consumption involving thermal variation removal in FinFET last-level cache and runtime voltage variation reduction via learning-based voltage emergency prediction; iii) a package-architecture co-design that addresses the pin-bandwidth pressure inside 3D packages and optimizes the performance and power efficiency of 3D heterogeneous multi-core systems via thread scheduling with respect to 3D package characteristics. Based on these contributions, this thesis underscores the importance of multi-physics co-design in 3D systems and highlights the value of these co-design practices as an integral part of future processor design.

CHAPTER 1
INTRODUCTION

The evolution of computer systems is progressing towards data-centric organizations driven by a combination of technology and application trends. Some major driving forces are i) the low spatial and temporal data locality and extensive data bandwidth requirements of emerging applications such as graph analytics, machine learning, relational computation, and neural-inspired learning algorithms, ii) the declining per-core pin bandwidth resulting from the slower growth of package pins compared to device counts, iii) increased energy consumption per operation because of the slowdown in technology scaling, and iv) the dominating cost of energy for data movement. To address the above challenges, researchers have resorted to a combination of three-dimensional (3D) packaging and through-silicon via (TSV) interconnects, which integrates the computing cores and memory modules in a single package, for large-scale, multi-core systems. Compared to the traditional 2D systems, such 3D integrated circuits (ICs) shorten the inter-tier wiring distance and promise a two order of magnitude increase in bandwidth between memory and compute nodes coupled with low energy data movement, providing a potential performance boost . However, the power density of 3D ICs will increase as a result of the breakdown of Dennard scaling in metal-oxide-semiconductor field-effect transistor (MOSFET) technology with the progression of Moore's law. Under such circumstances, the 3D multi-core processor encounters severe thermal challenges. Thus, to sustain performance scaling, we advocate the co-design of thermal management and cooling, power consumption, and power delivery in a 3D system, and emphasize the importance of improving system energy efficiency.

To ensure performance scaling under a power cap for 3D multi-core processors, we propose a multi-physics co-design paradigm that integrates the models of multiple simultaneous physical phenomena and their impacts on system performance into processor de-

sign practice. In our study, the elements of processor design include thermal management, power delivery and the package configuration. Specifically, the process consists of three major parts: i) characterizing the thermally dependent architecture performance under various cooling configurations, ii) analyzing the interaction between power management schemes and thermal effects, and studying its relationship with on-chip integrated voltage regulators (IVRs), and iii) investigating the co-design of architecture and package configurations that alleviate the effect of pin-bandwidth stress and optimize the processor configurations based on our analysis in memory hierarchy of 3D packages.

As such, this dissertation seeks to achieve high performance and energy efficiency in a 3D multi-core processor by adopting a multi-physics co-design methodology. Specifically, we focus on co-design practices of 3D processor design related to thermal cooling, power delivery, and package configurations. This dissertation is comprised of relevant publications [1] [2] [3] [4] [5] and research works, and the major contributions of this dissertation are summarized as follows.

The first contribution of this research is a thermal-architecture co-design of a processor that applies adaptation mechanisms that improve energy efficiency and performance relative to the worst case design, which is the state of the practice. We first characterize and model the thermal behavior of 3D ICs under alternative cooling configurations. The research explores co-design possibilities from two perspectives. First, to improve the performance and energy efficiency, we optimize microfluidic cooling structure with respect to architectural parameters such as the floorplan and the runtime power map and deploy computational sprinting of the cores with the optimized cooling. Second, we characterize the critical path delay of circuits in SRAM cells with respect to temperature changes, and then develop two adaptation mechanisms that convert runtime thermal headroom to performance. One mechanism is the reduced-cycle cache model (RCM), which changes the access latency of the SRAM last-level cache (LLC) as a function of the temperature of the cache bank. RCM improves the cache performance at a lower temperature over a cache

design based on the worst-case thermal behavior. The other mechanism is the partial boosting model (PBM). It uses the lower critical path delay at lower temperatures by applying dynamic voltage and frequency scaling (DVFS) to boost core frequency. It also introduces the LLC bank temperature as a negative feedback that prevents performance degradation caused by overheating.

The second contribution in this study is a co-design of the power delivery and power management components of the processor. The co-design reduces the runtime power consumption of the processor by minimizing the guardband of the supply voltage normally determined by the conventional worst-case analysis. In this research, we examine two subsystems of the processors: the cache and the core logic. For the cache system, we propose a constant performance model (CPM) that employs a voltage adaptive system inside the SRAM arrays of LLC. CPM reducess the thermal guardband of the supply voltage and increases the energy efficiency of the cache system over a wider operating voltage range. For the core logic, we apply a similar philosophy to the power management subsystem by examining the design of the power delivery network (PDN) and reducing the voltage guardband with the predictive data generated from the architectural models in the first part of the proposed work. Possible schemes of reducing the guardband include developing new functional circuits that compensate for the voltage droops inside the voltage regulator and producing architecture-level prediction of the worst-case voltage droop events that warm up the power system. These schemes optimize the required supply voltage that maintains the circuit timing constraints. To improve the system performance and energy efficiency, we also study the impact of various DVFS implementations on system availability and performance and explore novel transient architectures that continue to perform proper computations during the power transition.

The final contribution of this work is a package-architecture co-design approach that deals with the increasing pressure on pin bandwidth caused by the scarcity of data pins inside the package. We argue that a balance between the memory hierarchy and the comput-

ing cores is critical in the 3D system with respect to the optimization of energy efficiency. Toward this end, the work proposes a 3D architecture, called the Short-Stack, which is a two-stack configuration that uses the face-to-face bonding to couple a multi-core die and a cache die. The purpose of the Short-Stack is to reduce the package pin bandwidth pressure with a large LLC capacity implemented in low-cost 3D manufacturing technology. For example, we investigate cache implementations using both the traditional planar and fin field-effect transistor (FinFET) technology. In the second part of this contribution, we explore the design space of the core die across asymmetric cores (i.e., combinations of cores that span simple in-order cores to complex out-of-order cores accompanied by cache and network elements). The combinations differ in energy profiles and bandwidth demand, which also depend on the characteristics of the application running in the system. Understanding these relationships is critical to design effective thread scheduling policies in heterogeneous multi-core architecture of 3D processor-memory packages for energy efficiency

In summary, we wish to establish the value of the multi-physics co-design as an integral part of future processor design, in which we make the following contributions:

1. We establish the co-design viability of the 3D multi-core systems to address the power and thermal limitations and achieve the goal of high performance and energy efficiency.
2. We conduct a thorough architecture-level multi-physics characterization of 3D systems that includes thermal modeling, power analysis, and package study under the 2D, 2.5D, and 3D configurations.
3. We propose a thermal-architecture co-design to address the thermal challenges in 3D ICs, in which we do the following:

    - Co-optimize the geometric design of the microfluidic heat sink and power management based on core sprinting.
    - Develop two thermally adaptive mechanisms, RCM for the cache system and PBM for the core logic, both of which obtain high energy efficiency.

4. We introduce a power-architecture co-design that reduces the system power consumption by leveraging the supply voltage guardband. Specifically, we fulfill the following aspects of co-design:

   - Propose CPM for the LLC that reduce the thermal guardband of supply voltage over a wide temperature range.
   - Apply the architecture-level prediction of impending changes in power state to the voltage regulator, which reduces the runtime voltage droop.

5. We present a package-architecture co-design methodology that optimizes the performance and energy efficiency of heterogeneous multi-core systems with respect to 3D bandwidth characterization as follows:

   - Propose the Short-Stack 3D architecture that addresses the pin bandwidth challenges in multi-core systems.
   - Explore effective scheduling policies of 3D heterogeneous processors with respect to the refactored memory hierarchy in a 3D package.

The reminder of this dissertation is organized as follows. Chapter 2 summarizes related work encompassing 3D IC and package characterizations and processor co-design practice. We divide this chapter into three sections to demonstrate the work of processor design with thermal structure, power delivery system, and package configuration respectively. These previous studies lay the foundations for our research on multi-physics co-design of 3D processors.

Chapter 3 details the full-system simulation framework supporting multi-physics analysis. We present a proxy structure integrated inside a cycle-based simulation kernel to combine a multi-core emulation frontend with OS interaction, microarchitecture timing models, and a multi-physics library.

Chapter 4 presents our work in thermal-architecture co-design. We first characterize the thermal behaviors in a 3D IC and present the optimization of microfluidic pin fin based

on the floorplan and power map of a 3D processor. We then design a thermally adaptive cache in 3D processors to maximize performance and energy efficiency.

Chapter 5 demonstrates our research in power-architecture co-design. In the first part, we presents a power efficient cache based on the thermal-delay characterization in SRAM. The cache design is proposed to balance between the thermal headroom and voltage guard-band to minimize runtime power consumption. In the second part, we propose a voltage emergency prediction mechanism based on architectural information inside an on-chip voltage regulator integrated into a 3D processor to reduce voltage variations during load transients and reduce power.

Chapter 6 discusses the opportunities in 3D IC packaging. First, we review the pin stress problem and propose a 2-tier 3D processor using an eDRAM LLC to minimize off-chip data traffic. Second, we design a heterogeneous 3D processor combining in-order and out-of-order cores and implement an efficient thread scheduling algorithm for heterogeneous processors to maximize performance within given power and thermal cap based on the characterization of the refactored memory hierarchy in 3D systems.

Based on the three levels of co-design practice in a 3D multi-core processor between microarchitecture, thermal management, power delivery, and package interaction, this dissertation wishes to underscore the importance of multi-physics co-design in achieving high performance and energy efficient for future processors.

CHAPTER 2
RELATED WORK

In this chapter, we summarize prior work related to processor co-design in terms of thermal structure, power delivery system, and package characteristics.

2.1 Thermal-Architecture Co-optimization in 3D Systems

Due to the failure of the Dennard scaling in MOSFET devices, multi-core processors are limited by power regardless of chip organization and topology [6]. Thus, processor designers emphasize system energy efficiency to sustain the scaling in multi-core systems, in which the 3D IC technology becomes promising to address the problem [7]. The 3D IC overcomes the limit of off-chip interconnects to provide low latency, high bandwidth, and low energy-per-bit via the vertical TSVs, as this vertical integration could reduce the global wire length and power by 50% [8] and increase wire-limited clock frequency nearly four-fold [9]. However, as the integration level continues to increase in the 3D IC, the thermal problem arises from several factors detailed in [10]. Specifically, the non-uniformity of the heat dissipation would increase heat fluxes of the hotspot by ten times [11].

To dissipate the excessive heat in 3D ICs, several researches have resorted to the liquid cooling using pin fin enhanced micro-gaps as a viable solution. Zhang et al. [12] fabricates an inter-tier pin fin enhanced micro-gap and shows that a staggered pin-fin heat sink is able to provide a thermal resistance as low as $0.27K \cdot cm^2/W$. Jasperson et al. [13] compares micro-pin fin and micro-channel heat sinks. Their results show that micro-pin fin heat sink has a lower convection thermal resistance at liquid flow rates above approximately $60g/min$, with a higher pressure drop. Ndao et al. [14] finds the maximum temperature in a chip could be maintained at $56°C$. Moreover, the paper by Bejan and Morega [15] reports the optimal geometry of an array of fins that minimizes the thermal resistance between

the substrate and the flow forced through the fins. These researches suggest microfluidic cooling as a viable solution for 3D chips, but their work primary focus on minimizing the thermal resistance at the mechanical level, and a co-optimization between heat sink, ICs, and processor architectures can achieve better runtime thermal properties as follow.

Because of the non-uniform heat determined by floorplan in the processor, the co-design of the architecture floorplan and thermal structure is critical to improve the thermal dissipation capability and reduce the thermal-related leakage power. [16]. Sarvey et al. [17] examines five configurations of the micropin-fin arrays in memory-processor stacks, indicating that the stack arrangement of the memory and core dies affects the hotspot temperature of the chip. Kidd Chen et al. [18] proposes a co-design methodology in system-on-chip between chip, package, and PCB that reduces the thermal resistance of the bumping structure in system-on-chip. Wang et al. [16] minimizes the thermal resistance of the micropin-fin structure providing the runtime power map of the 3D processor. The philosophy under these researches seeks to perform a holistic optimization between system organization and thermal structure with circuit- and system-level information.

Apart from the optimization work on the thermal resistance in the 3D system, other system researchers seek to improve the performance and energy-efficiency of the multi-core processor by utilizing the thermal characteristics of the 3D chips. John et al. [19] designs a temperature-aware subarrys in the cache system that minimizes the leakage power. Jia et al. [20] optimizes the workload data in the embedded system with hybrid memory based on the temperature distribution to achieve optimal performance and temperature. Wang et al. [21] applies a thermal-aware task scheduling policy on the multi-core processors to reduce the energy consumption. These researches utilize the thermal data as a feedback/guideline to the system-level functional units for power reduction and performance improvement, and indicate the potential benefits of a thermal-architecture co-design to 3D processors.

2.2 Power-Architecture Co-design in 3D Systems

A effective way of power reduction is to reduce the supply voltage [22]. This is achieved at the risk of violating the timing constraints of the very-large-scale integration (VLSI) circuits [23]. To avoid this problem, researchers investigate the power delivery network (PDN) of the 3D chips to interpret the timing behavior of the 3D IC. A PDN models the IR drop of the circuit in a chip, depicted as a low-pass filter with RL segments in series attached with capacitors at each end [24]. Lee Young-Joon et al. [25] reports the timing analysis of a many-tier 3D IC, and proposes timing optimizations that increase the clock frequency. He Huanyu et al. [26] observes the distinct impact of PDN in 3D integration with a detailed SPICE model. The discovery from these researches serves as the background of PDN developed in our experiments.

Another critical component of the power subsystem in 3D ICs is the voltage regulator. The implementations of voltage regulators include a linear regulator and a switching mode power supply. The switching power supply outperforms the linear in terms of efficiency [27], and are widely used in high-frequency circuits such as the modern processors. In fact, the voltage regulator module of 3D processors often take advantages of the buck converter (one type of the switching regulator) that minimizes the power losses. Kohei et al. [28] dicusses the 3D buck regulators in the research of stacked-chip. More recently, Sun et al. [29] explores the potential of using 3D integration of the buck converter in 2D BiCMOS technology, and Sergio et al. [30] studies the DC-DC converter as a separate die for high performance PDN by utilizing a voltage module with on-chip voltage regulators. Furthermore, on-chip voltage regulators need to address the voltage droop properly. Li et al. [31] and Song et al. [32] work on the floorplan and physical design to reduce voltage droops by inserting decoupling capacitors. These researches present a direction to improve the effectiveness of voltage regulator in 3D ICs and minimize runtime voltage emergencies.

Besides applying the methods at the circuit level, other researchers from other regime

share insights through a co-design methodology. Berthiaume [33] describes the effect of voltage droop on the timing behaviors of the processor. Vosicher et al. [34] applies the hysteretic controller with cycle-by-cycle current limiting to the computer processors to improve the transient response affected by the voltage droop. Hu et al. [35] looks at the voltage droops caused by runtime applications, proposing a layer-independent scheme to balance the intra-layer voltage droop via OS scheduling. Vijay et al. [36] instead implements a voltage emergency predictor to prevent worst-case droops and surges. This signature-based predictor predicts the hazard microarchitectual events, allowing the processor to operate with tight voltage margins (4% compared to 13%). Leng et al. [37] unveils the program dependent V_{min} across graphics processing unit (GPU) programs based on a kernel's microarchitectual performance counters to improve the overall energy efficiency. Kim et al. [38] targets at the floating-point unit with a throttling technique that translates the relaxed voltage margin to performance improvement. These work show power reduction with power-aware schemes in the processor.

In addition, a commonly-used power management technique in modern processors is dynamic voltage and frequency scaling (DVFS), which dynamically manipulates the voltage and frequency of a component, depending upon circumstances. DVFS improves the power efficiency of the processor. Several studies dig into the problem to optimize the implementation schemes of DVFS. Won Jae Yeon [39] proposes a per-core DVFS technique inspired by the congestion control protocol called TCP Vegas for the uncore shared resources to reduce the total energy dissipation. Torng et al. [40] integrates the asymmetry-aware work-stealing mechanism to the switching regulators that enable fine-grain DVFS per-core. Prasanthi et al. [41] proposes a control loop in the buck converter by monitoring the critical path delay for the adaptive voltage scaling. From the researches, we conclude a vital relationship between DVFS and voltage regulators, which promotes a co-design necessity in the power system of processors.

2.3 Package-Architecture Co-design for 3D Processors

As the processor moves to extreme scales, the IC package experiences an increasing pressure on the off-chip pin bandwidth. According to a recent study by Phillip et al [42], supply pins will take a large proportion of the total available pins in the package.

To overcome the drawbacks of the 2D packages, advanced packaging technology has emerged recently for the processors. For example, AMD puts the high bandwidth memory (HBM) on top of its Fury GPU in a 3D package connected through TSV, increasing the in-package bandwidth to above $100GB/s$ per stack [43], while Intel releases the embedded multi-die interconnect bridge, providing a low power, high bandwidth interconnect as a 2.5D packaging [44]. These advanced techniques not only enable the in-package integration of the main memory that reduces the off-chip bandwidth demands, but also increase the intra-tier bandwidth that facilitates heterogeneous integration. As a result, processor designers can explore heterogeneous designs of computer architectures in a single package to obtain high performance and energy efficiency. For example, various researchers [45] [46] [47] explore processing in memory (PIM) from the concept of near-memory computing, to get performance improvement with acceleration in memory requests. The extra logic layer in PIM are enabled by 3D packaging technology, and needs a carefully arrangement in the chip stack. Alternatively, Kim et al. proposes a 3D logic-memory system, Neurocube [48], for deep neural networks that process big-data applications.

Meanwhile, other scientists [49] [50] look into heterogeneous processor design that integrates multiple core types together to optimize performance, energy, and efficiency. The combination of core types and the utilization of such heterogeneous design is important to performance scaling of future processors. These researches demonstrate the importance of heterogeneous co-design in 3D systems for processor designers to obtain high performance gain and energy efficiency.

CHAPTER 3
SIMULATION INFRASTRUCTURE

3.1 Introduction

Modeling and simulation of multi-core systems is an important technology for future processor design. In our research, we extend the use of a cycle accurate full system simulator called Manifold [1] to perform $holistic$ analysis of system performance, power, energy and thermal concerns and to enable advanced processor management such as workload allocation and dynamic frequency scaling (DFS) for processor design in a flexible manner, which is done through an extensible unified interface built upon the MPI library called ManifoldProxy detailed as follows.

3.2 Simulation Model Overview

The Manifold simulation framework consists of multiple layers depicted in Figure 3.1. Two major layers are the $kernel$ and $models$ layers [1]. The $kernel$ layer provides parallel simulation services such as message passing and synchronization and the $models$ layer is a collection of microarchitecture components built on top of the simulation kernel. The frontend of Manifold uses QSim [51] multi-core emulator frontend to generate instruction flows to drive the simulation with a thread-safe callback-based application binary interface (API). To support physical simulation and analysis, Manifold embeds a structure of performance counters in the timing model of microarchitecture components to record pipeline activities at each sampling intervals, used as inputs of a multi-physics modeling framework named KitFox [52].

The structure of the system simulator is demonstrated in Figure 3.2. The core model consists of two pipeline configurations including both an in-order and out-of-order core

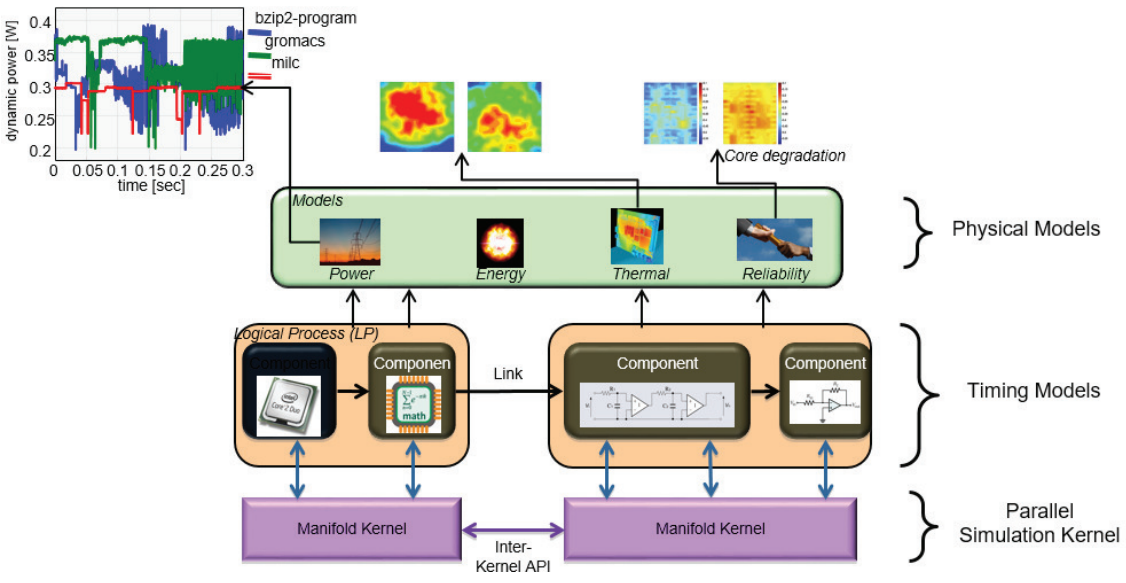


Figure 3.1: Overview of the Manifold simulation framework [53]

design derived from commercial processors. The cache model of the system employs the MCP-cache coherence framework, implementing a directory-based coherence of between private L1 caches and the shared LLC. The interconnection model utilizes Manifolds IRIS model with interfaces and routers. In our simulation, the routers are connected in a two-dimensional torus, while the network interfaces connects to the LLC banks and memory controllers. The main memory is modeled as a cycle-based memory controller and multi-bank DRAM array.

To facilitate micro control in processor management, a system monitor is constructed inside the simulator. The monitor coordinates the execution of processors and the analysis of physical models to achieve high-performance and energy efficiency, as detailed in the following chapters.

3.3 ManifoldProxy Interface

When the simulator is initialized, components are connected through the input and output ports for communication. Each input is assigned to an event handler that deals with in-

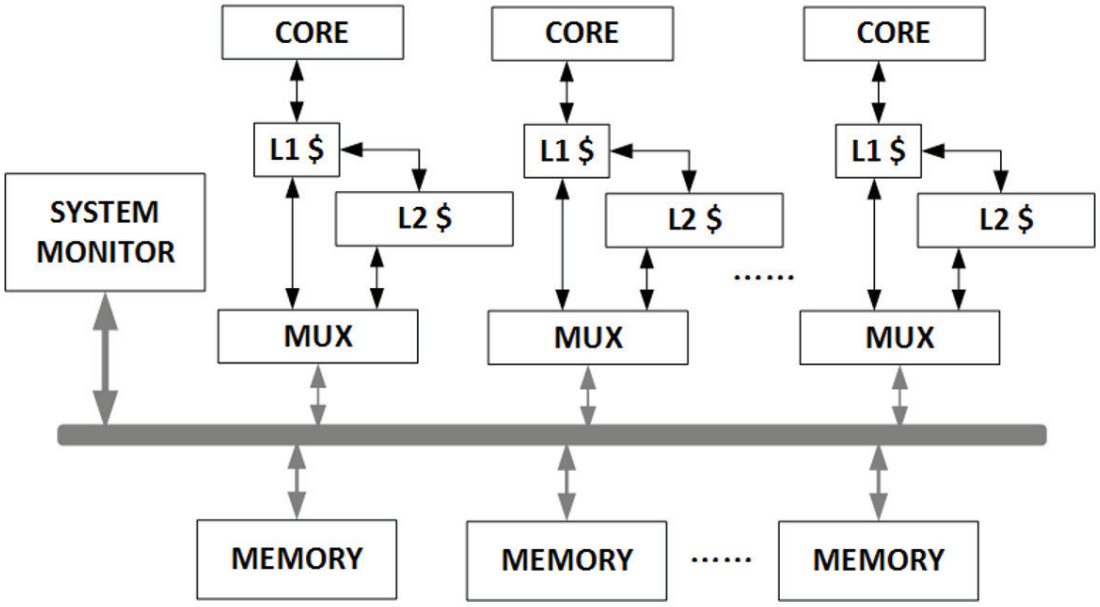


Figure 3.2: The simulator structure for multi-core processor simulations

coming data; each output inherits the $Send$ function from the simulation kernel to send data to its connected input port(s). One restriction of this $Send$-and-$Receive$ mechanism in Manifold is that connections are statically defined and cannot be changed in the middle of a simulation experiment, which create challenges for complex processor management. For example, thread migration is difficult as the QSim frontend is deeply coupled with the processor core model via callback functions provided by the QSim APIs.

To address the above problems, we extend the communication mechanism in Manifold and propose a proxy-based structure called ManifoldProxy as a standardized interface for complex interactions between components. ManifoldProxy is a dedicate Manifold component designed for communication, which is derived from Manifold $component$ class. It wrappers the $Send$ function and event handler as its members functions and implements an algorithm to manipulate comprehensive data and control triggered by a dedicate clock. With ManifoldProxy, Manifold can deploy external simulation libraries and software as Manifold plugins that enrich the features of microarchitectural analysis.

Each ManifoldProxy instance consists of a manager-client pair. The $manager$ subcomponent provides services that talk directly to the plugin software; the $client$ subcomponent sends data requests and control signals to the manger when invoked by other Manifold components (i.e., microarchitecture models). In sum, ManifoldProxy adds following features on top of existing implementation:

- Improving Scalability:

  ManifoldProxy decouples QSim and KitFox from Manifold source code to detachable plugins that are encapsulated into a ManifoldProxy manager submodule. Since the external library and software is instantiated to a separate MPI rank, ManifoldProxy can execute Manifold and plugins in parallel via MPI implementations.

- Extending Message Types:

  Initially, connection between two components only supports a single message type statically defined in a Manifold simulator. Therefore, when multiple types of messages need to be transferred within the two components, Manifold has to instantiate multiple links explicitly during the simulator setup, adding complexities in implementation and maintenance. In ManifoldProxy, incoming messages are first handled by a pre-processing unit so that multiple messages types can be supported through a single ManifoldProxy connection.

- Supporting Various Linking Topology:

  ManifoldProxy provides one-to-many and many-to-many mapping s between component instances. The manager submodule maintains a centralized buffer to deal with messages sent to or received from multiple directions, which adds flexibility for the plugin software to communicate with several microarchitectural components simultaneously. For example, the thermal library requires all power models to be ready before the temperature calculation. In this case, the thermal library connects with these power models through a single ManifoldProxy link.

- Implementing Dynamic Links:

  ManifoldProxy supports dynamic links between components during runtime simulation for one-to-many and many-to-many links. By sending a control command to the server submodule, ManifoldProxy can change the mapping topology between several instances, which can mimic complex runtime managements such as thread migrations.

We detail two proxy implementations QsimProxy and KitfoxProxy based on ManifoldProxy for QSim emulator and Kitfox library deployed in our multi-physics simulation and analysis.

3.4 Thread Management

Manifold uses QSim as the simulation frontend executing a guest consisting of a lightly-modified Linux kernel and a benchmark application [1]. The guest can be viewed a set of virtual CPU threads. When the run function is called, QSim generates callbacks to pass the instruction flow from the guest environment back to the host simulator for instrumentation such as virtual addresses and instruction operands. QSim supports a variety of benchmarks including general shared memory benchmarks such as SPLASH-2 [54] and PARSEC [55], and graph-computing benchmarks such as GraphBIG [56].

QsimProxy connects the QSim threads and instances of core model as shown in Figure 3.3. QSim resides in a server submodule that generates the instruction flows; core models reside in a client submodule that fetches instructions. The client submodule maintains a buffer to store instruction flows encapsulated in the $QueueItem$ structure. When the client buffer of a core is below a pre-defined threshold, it sends a data request to the QSim server submodule to feed the buffer by calling the run function. The data request contains the thread information so that QSim can push the corresponding instruction callbacks to the required $QueueItem$ buffer. In a homogeneous processor design, each core is associated with an identical thread and thus the core id is also used as the thread id in QSim.

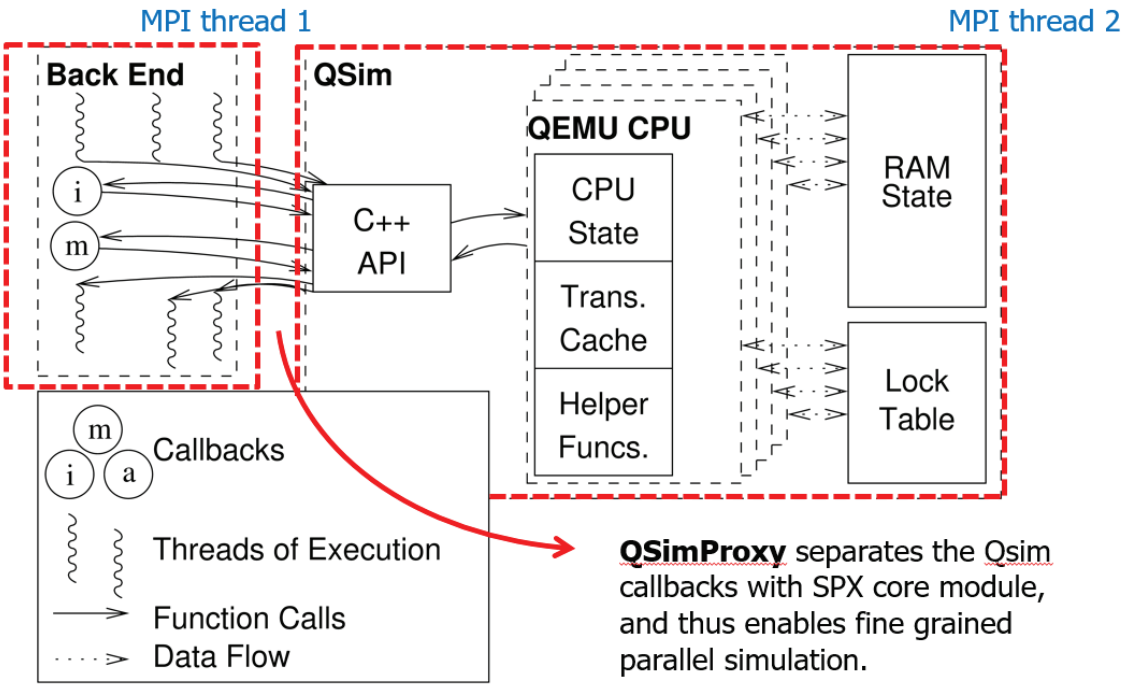


Figure 3.3: QsimProxy using QSim emulator as the simulation frontend

3.4.1 Thread Migration

In QsimProxy, a QSim thread is associated with a core instance through the thread id. Such assignment is done when the core instance is initialized and remains unchanged throughout the simulation. To support thread migration from one to the other core, QsimProxy adds a control command to swap the executing thread to a different core by changing the thread id dynamically during execution. In our simulation framework, thread migration is controlled by a centralized monitor that tracks the runtime metrics of the processor, as shown in Figure 3.4 (a). Migration overhead is modeled as the miss penalty of private L1 caches.

3.4.2 Thread Partition

Supported QSim benchmarks are partitioned into symmetric multi-threads with similar runtime behaviors. Therefore, when each core is assigned to a single thread, this scheme of thread partitioning is not efficient in a heterogeneous processor, as both the in-order and

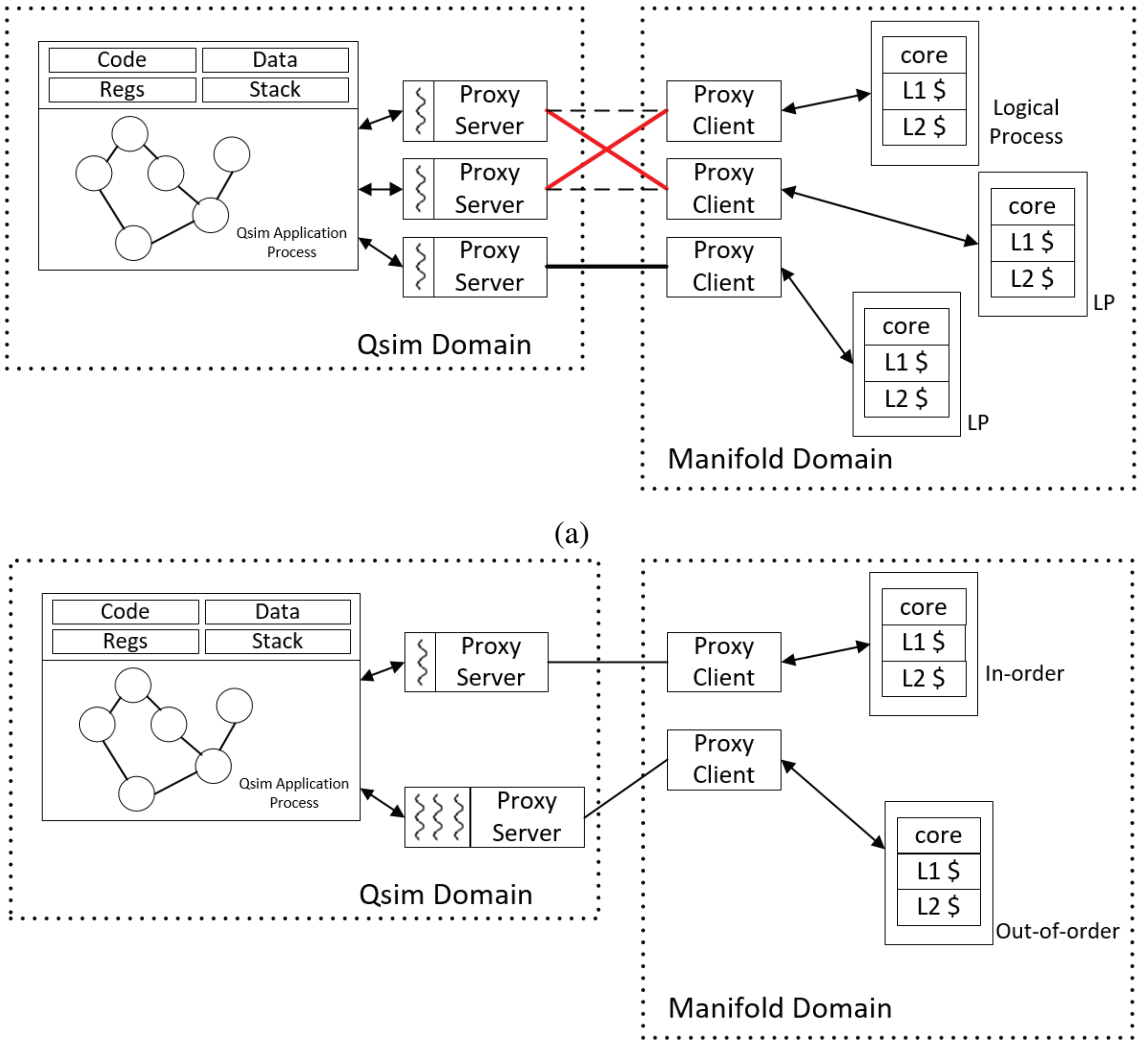


Figure 3.4: QsimProxy execution model: (a) thread migration (red), and (b) thread partition

out-of-order cores are given similar amount of workload.

Instead of modifying the guest OS, We are motivated by simultaneous multi-threading processor for out-of-order cores, in which a single pipeline can execute instructions from different threads at the same cycle. We define a variable $threadNum$ for out-of-order cores to specify the number of threads running on an out-of-order core and implements a many-to-many topology in the ManifoldProxy connection between QSim threads and out-of-order cores, as shown in Figure 3.4 (b). The QsimProxy in the heterogeneous design also supports thread migration as described in the previous subsection.

3.5 Multi-physics Analysis

Figure 3.5 demonstrates the integration of the multi-physics library KitFox into Manifold using KitfoxProxy via an one-to-many mapping. KitFox is wrapped in a client submodule that sends data requests of performance counters to Manifold timing models periodically by a clock with the frequency set to the sampling rate. ManifoldProxy forwards the counter information to the corresponding power model configured by KitFox to calculate runtime power consumption of microarchitectural components. We use a queue structure to synchronize the power information. When the power of all components are updated, KitfoxProxy invokes the temperature computation through KitFox APIs for thermal analysis. Outputs of KitFox are redirected to a system-level monitor for controlling purposes and a statistics module for printouts.

One important feature of the KitfoxProxy is that the KitFox client utilizes a non-blocking mechanism to handle the multi-physics computation. Performance counter retrieval, power calculation, and thermal analysis are divided into three separate processes that do not block each other. As a result, KitfoxProxy improves the parallelism of the Manifold simulation process.

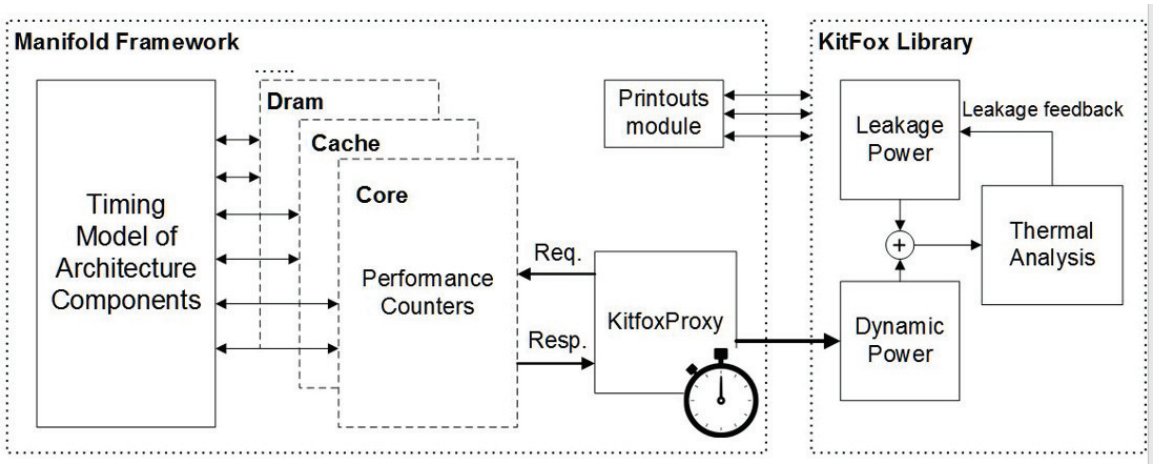


Figure 3.5: KitfoxProxy connecting the processor timing model with the KitFox multi-physics library

3.6 Dynamic Frequency Scaling

In Manifold, clocked components defines two functions, $rising$ and $falling$, to response to a clock event (i.e., rising/falling edge). The internal scheduling module calculates the absolute time of the next clock tick by iterating over the $nextTickTime$ function of clocks and wakes up the event with the smallest absolute time. Initially, Manifold maintains a $time$ variable that records the current time of the simulation accumulated by the clock period each time $nextTickTime$ is called. The truncation error of the clock period (calculated as $1/freq$) can be ignored if the frequency of clocks does not change throughout the entire simulation.

However, in situations that a clock frequency constantly changes, the truncation error could ultimately lead to scheduling distortion as the error gets accumulated each tick. To solve this problem, we introduce the $m\_lastChangeTick$ and $m\_lastChangeTime$ variables in $nextTickTime$ that record the tick numbers and time right before the frequency change. Since the two variables checkpoint the previous timing information, Manifold can schedule clock events as if the frequency never changes. We present a comparison of the two algorithms as follows.

Algorithm 1 Comparison of simulation time calculation

```
1: procedure NEXTTICKTIME
   ▷ Constant Clock Frequency
2:     time += 1 / freq;
   ▷ Dynamic Frequency Scaling
3:     time = (nextTick - m_lastChangeTick) / freq + m_lastChangeTime);
```

Algorithm 2 Clock frequency update

```
1: procedure SETFREQUENCY(f)
2:     if nextTick != m_lastChangeTick then
3:         m_lastChangeTime += (nextTick - m_lastChangeTick) / freq;
4:         m_lastChangeTick = nextTick;
5:         freq = f;
6:     else
7:         throw MultipleFreqChangeException;
8:     end if
```

Moreover, we restrict Manifold to allow only one frequency change with in a clock tick by comparing the current clock tick number with $m\_lastChangeTick$ to prevent racing conditions in the event scheduling. When multiple frequency changes happen within a single tick, the clock object throws an exception.

3.7 Summary

In this chapter, we extend a cycle-based simulation framework Manifold with a proxy-based structure ManifoldProxy to support multi-physics analysis and comprehensive processor managements such as dynamic voltage scaling and thread migration. The proposed ManifoldProxy utilizes the MPI parallel library in complex communication between Manifold timing models and external (plugin) libraries and software. Compared to the original design in Manifold, ManifoldProxy improves the parallelism of microarchitectural simulation using a non-blocking communication mechanism between components and increases the flexibility of constructing a processor simulator with supports in various connection topology and dynamic linking.

CHAPTER 4

CO-DESIGN OF PROCESSOR ARCHITECTURE AND THERMAL COOLING

4.1 Introduction

The cooling system in a processor consists of the internal and external cooling modules [57]. The internal cooling module transfers the heat from the inside chip to external heatsink, while the external module exchanges the heat between the package and ambient atmosphere. Figure 4.1 depicts the two cooling modules in a single-socket package.

The internal module, in which the heat is dissipated through conduction, includes the printed circuit board (PCB), substrate, flip-chip, thermal interface material (TIM), and heat spreader (lid). The external cooling serves as the means to transfer the package heat out to the environment. In this chapter, we focus on two types of cooling techniques detailed as follows.

Air Cooling

The typical heat sink for the air cooling is shown in Figure 4.2. It contains a base region that contacts to the package and a fin region to extend the surface area for heat transfer. The thermal resistance of the heat sink is determined by many factors, including fin thickness, fin height and air flow rate. Specifically, the thermal resistance of the straight-fin heat sink decreases when the fin height and fin numbers increases, indicating a better thermal performance. The air convection cooling we model is a copper plate straight-fin heatsink [58] of $8.55cm$ x $6.85cm$ in dimension with 40 fins of $3.4cm$ in height.

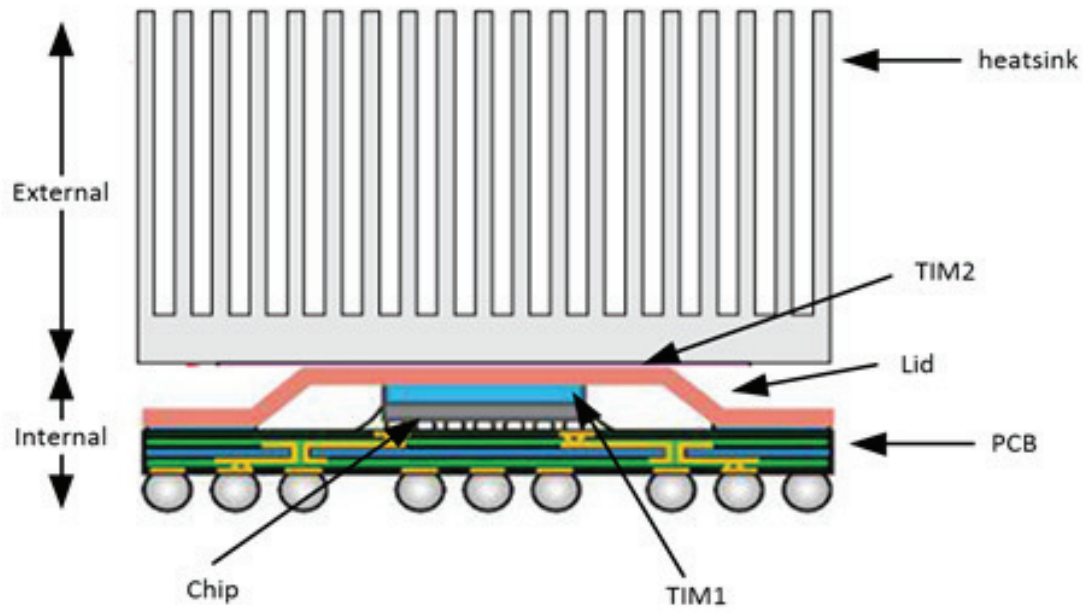


Figure 4.1: A typical cooling system for a single-chip package

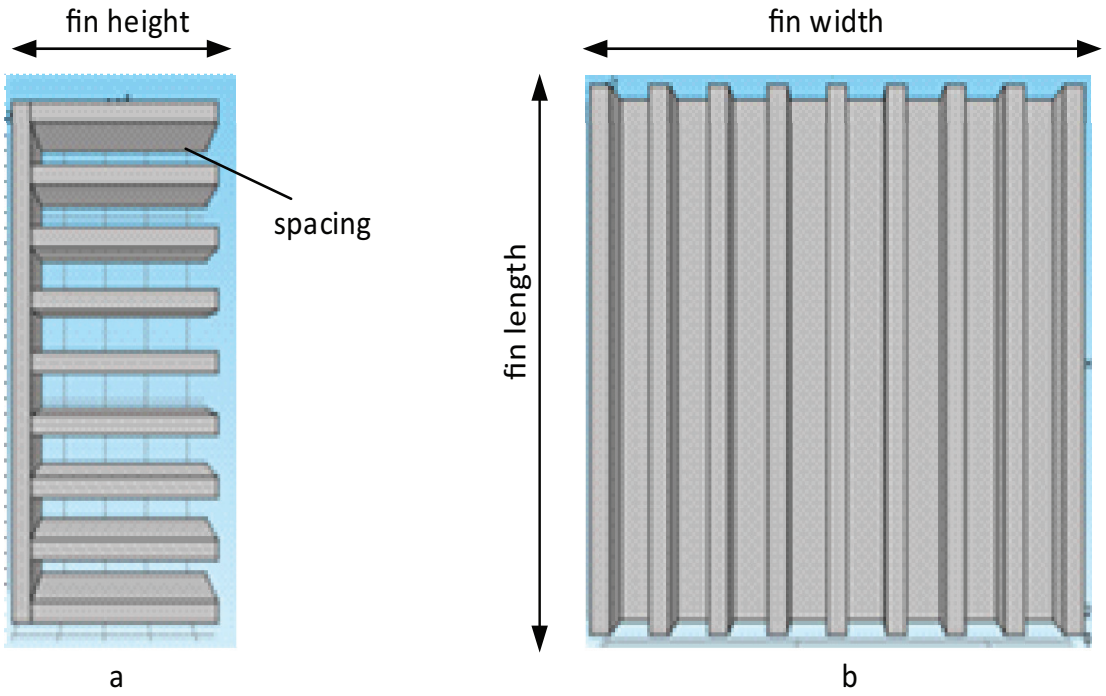


Figure 4.2: Straight-fin heatsink for forced air cooling

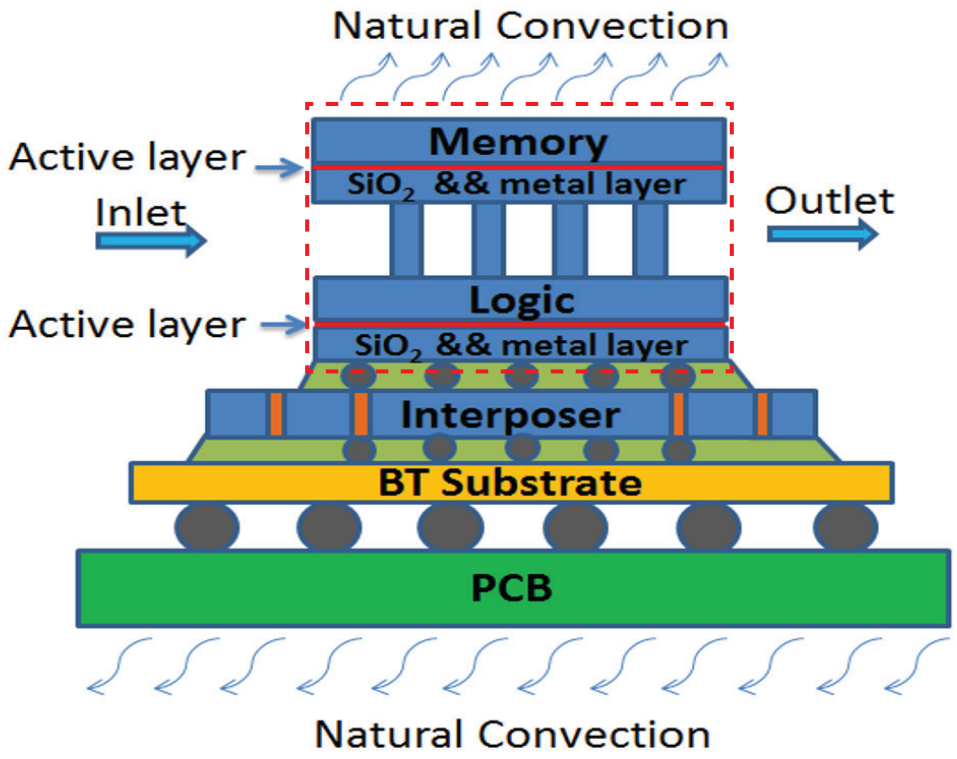


Figure 4.3: Micro-pin fin heatsink for single-phase microfluidic cooling [16]

Microfluidic Cooling

As the straight-fin heatsink has limited cooling capabilities in advanced packaging such as 3D ICs, researchers look for alternatives in these situations and demonstrate the feasibility of inter-tier microfluidic cooling. Figure 4.3 shows one common microfluidic structure, the micro-pin fins using single-phase cooling. The pin fins are embedded between tiers and the fluid goes through them from the inlet to the outlet and removes heat. Due to the much higher heat capacity of liquid (e.g., water) than air, the cooling capacity of microfluidic cooling surpasses that of air cooling. For a two-tier chip, microfluidic cooling has been shown to handle a total power dissipation of $200W$ [59].

4.2 Microfludic Cooling Co-Design and Leakage Power Minimization in 3D Stacked Multi-Core Chips

4.2.1 Motivation

The 3D stacked IC integrates multiple dies vertically in a single package and provides high integration density. Compared to a 2D planar design, it shortens the die-to-die distance and substantially increases the inter-die communication bandwidth. However, the high package density in 3D ICs results in higher power density per unit volume of the package and exposes challenges for thermal management.

Furthermore, leakage power of ICs becomes a major concern in sub-32nm technology following the end of Dennard scaling. For example, studies show that for a 2-way cache in $16nm$ node with the size of $2MB$, an estimation of leakage power could be up to $8W$ at $60°C$ [60]. The increased power densities of 3D packages exacerbate the temperature-leakage coupling making the situation even worse. Liquid cooling with surface enhancements such as micro-pin fin is a viable solution to address the thermal problems in 3D processors because of its much better thermal dissipation capability compared to conventional cooling techniques. Together, both phenomena reduce energy efficiency and requires processor designers to optimize the cooling structure in 3D packages.

Previous work involves optimizing pin fin configurations to achieve low thermal resistance under a static power density and a given processor floorplan [61]. We argue that the energy efficiency of 3D processors can be further improved if we co-optimize pin fin structure and runtime power concurrently. We investigate the co-design between processor floorplans and inter-tier pin fin designs and quantify the improvements in system throughput and energy efficiency with the optimized pin fin cooling.

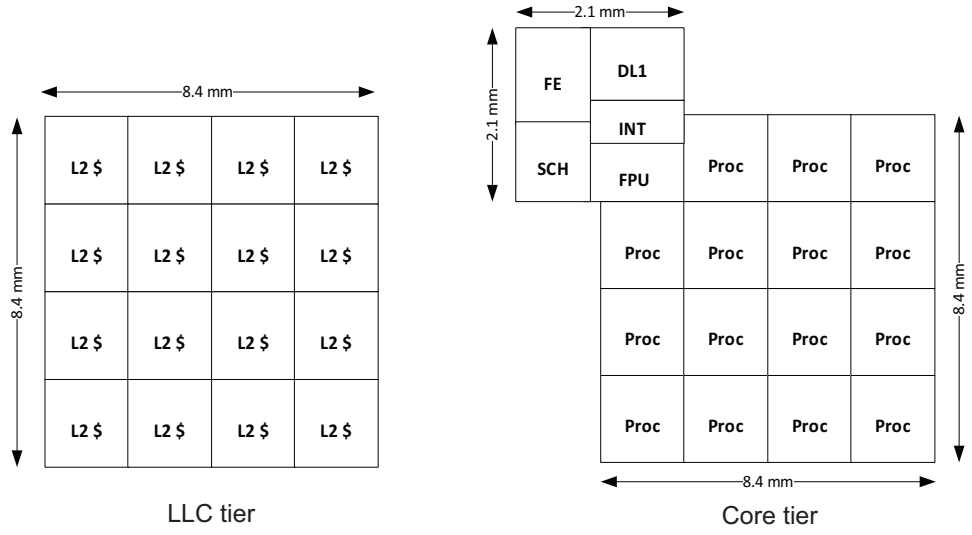


Figure 4.4: Floorplan of a 2-tier stacked processor

4.2.2 Thermal Characterization in 3D ICs

We characterize the impact of microfluidic cooling on leakage power in terms of pin fin configurations based on a compact thermal model for a 2-tier 3D processor shown in Figure A.1. The thermal model is first introduced by Zhimin Wang [16] and we present the modeling details in Appendix A.

The 3D processor consists of a separate core and LLC tier, electrically connected by Through Silicon Vias (TSVs). The core tier is placed at the bottom closer to the package primarily for power delivery considerations. Cores resemble the Intel Nehalem architecture with a private L1 data cache of $128KB$ and a shared LLC. The LLC tier includes 16 SRAM banks with a per-bank size of $2MB$ and connects to external DRAM controllers via a 2D torus interconnection. The 3D 16-core processor is modeled in $16nm$ technology with a floorplan dimension of $8.4mm \times 8.4mm$, as depicted in Figure 4.4. Pin fin arrays are constructed between the two tiers dissipating heat from both processors and caches.

We extend the circuit models in McPAT [62] with thermal effect from two perspectives. Firstly, we update the feedback loop of temperature and leakage in $16nm$ technology based on our HSPICE model using device parameters based on ITRS report 2007 [63]. We con-

struct the temperature-leakage curve of a unit $16nm$ FinFET device and record its leakage current at discrete temperature points from $300°K$ to $400°K$ in a lookup table. To support a continuous temperature range, we deploy a linear interpolation for leakage current calculation between two sampling points.

Furthermore, we model the thermal effect of FinFET devices on the mobility and threshold voltage with respect to temperature variations and update the CACTI model [64] for caches. A temperature-related term is appended to calculate the global V_{Th}, switch-on current I_{on}, and sheet resistance R_per_um respectively. The updated model can also evaluate the thermal impact on transistor speed for temperature-delay analysis.

4.2.3 The Co-Optimization Framework for Micro-pin fins

By assuming linear temperature variations within the silicon base and silicon dioxide layers, we rewrite the energy equations in the compact thermal model as follows:

$$\begin{aligned} Silicon\ base : T_{base} &= k_1 \cdot z + a, \\ SiO2 : T_0 &= k_2 \cdot z + b, \\ Pin\ fins : T_{fin} &= T_{f,in} + C_1 \cdot e^{mz} + C_2 \cdot e^{-mz}, \end{aligned} \tag{4.1}$$

where k_1, k_2, a, b, C_1 and C_2 are the model constants defined by boundary conditions. In our model, these constants are expressed as functions of temperatures of coolant, cores, and LLC. We utilize the tridiagonal matrix algorithm (TDMA) to calculate the temperature field.

We perform the geometry optimization of the pin fin structure offline based on the power traces from architecture simulations. Specifically, we embed the thermal model and processor floorplan into an optimization framework using the genetic algorithm provided by MATLAB. As shown in Figure 4.5, the key elements are the setting of objectives and constrains. In our experiments, we set the objective to find the pin fin dimensions that produce minimum junction temperature under a certain power map and a fixed pumping

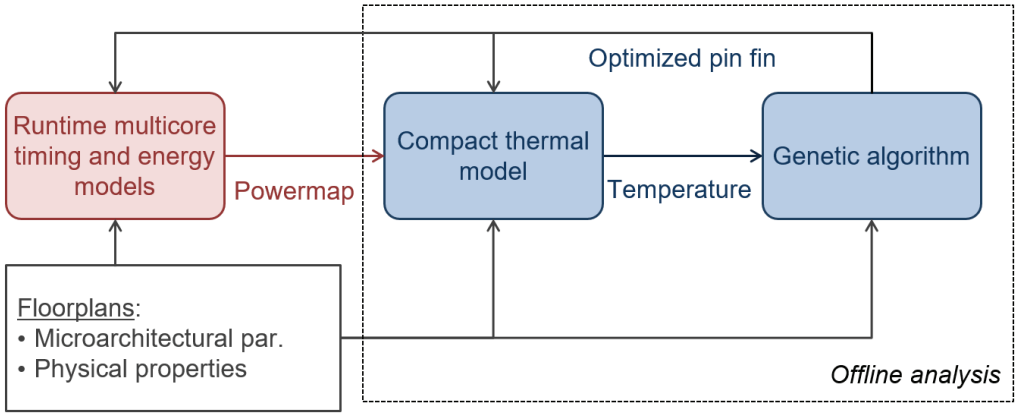


Figure 4.5: The optimization process of pin fin structure

Table 4.1: Time and space complexity of barnes and ocean-c

Application	**Time Complexity**	**Space Complexity**
barnes	$NlogN$	N
ocean-c	N^3	N^2

power of $0.03W$. During the optimization, the genetic algorithm would first generate randomly individual dimensions to input to the compact model as possible solutions to the optimization. Then the temperature field and junction temperature are determined for each solution. Individual solutions are later selected through a fitness-based process, where fitter solutions are typically more likely to be selected as parent solutions. Child solutions are produced using the method of crossover and mutation based on the parent solution. The new temperature field and junction temperature are computed for each new solution with maximum generation set to 100. The stopping criterion is that the function tolerance is less than 1e-6.

4.2.4 Results and Analysis

We choose two applications $barnes$ and $ocean$-c in the SPLASH-2 benchmark for evaluation. $Barnes$ is a typical computational-bounded application as it has low cache miss rate; $ocean$-c is a memory-bounded application due to its relatively high miss rate and large remote traffic. The time and space requirements [55] are listed in Table 4.1.

Table 4.2: Configuration parameters of the micro-pin fin structure

	Diameter (μm)	**Pitch Spacing** (μm)	**Height** (μm)
baseline (bsl)	100	200	200
optimized (opt)	180	320	400

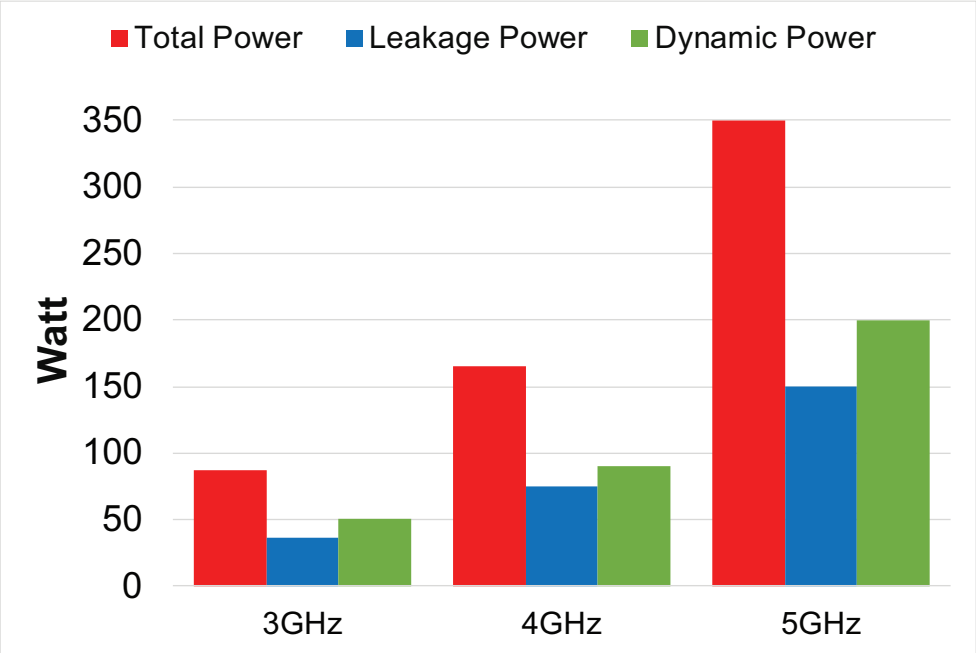
The baseline cooling configuration is a micro-pin fin structure. One major advantage of using it over the conventional air cooling is the much less power needed by the cooling system to achieve same thermal resistance. Because the fan power is a cubic function of air velocity [65], increasing the air flow rate (and thus the heat transfer coefficient) in the straight-pin heatsink would dramatically increase the fan power.

With respect to comparison with air cooling there are several points to note. First, a head-to-head comparison is difficult since thermal operating regions with microfluidics are considerably higher than that which can be achieved with air cooling. Second, to achieve comparable thermal behaviors, the power consumption of the fans in an air cooled implementation will be substantial one to two orders of magnitude more than the pumping power expended in microfluidics. Thus, the bulk of the power and energy advantages of microfluidic cooling compared to air cooling is derived from the reductions in the relative power expended in the cooling system.

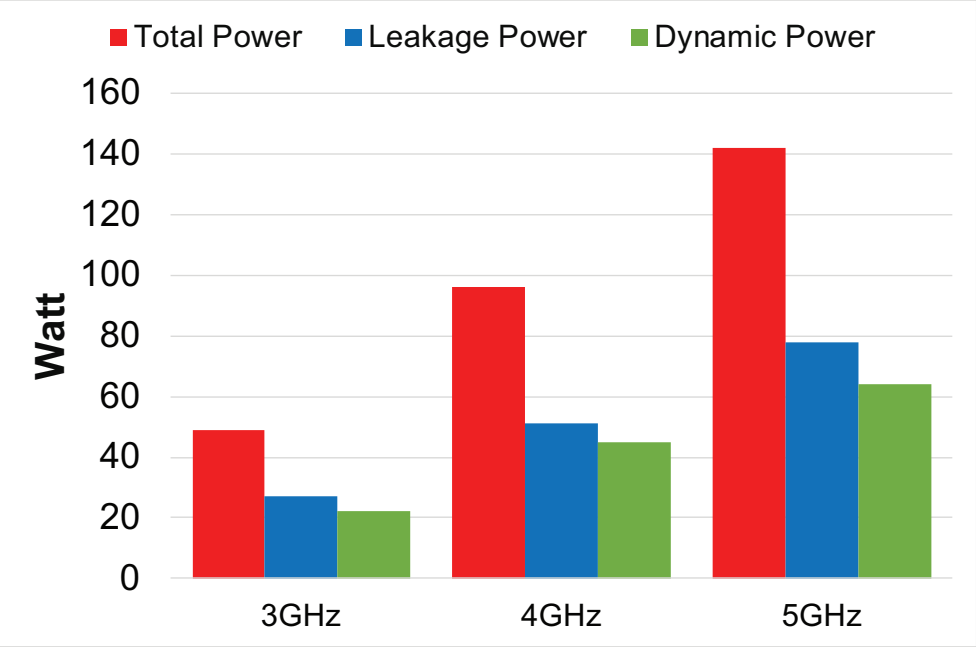
We put the runtime power map into the co-design framework and optimize the geometries of pin fins based on the baseline configuration. Table 4.2 shows the configuration parameters between the baseline and optimized pin fin.

We first evaluate the power consumption with respect to the processor frequency. Figure 4.6 indicates that both the dynamic and leakage power follow a quadratic relationship to the system frequency. To support higher clock frequency, the system supply voltage needs to be scaled up accordingly, and the leakage power will take a larger proportion of total power consumption in a higher system frequency. Notice that the leakage power in memory-bounded application $ocean\text{-}c$ takes up over 50% of total power consumption.

To better understand the relationship between cooling parameters and processor power,



(a)



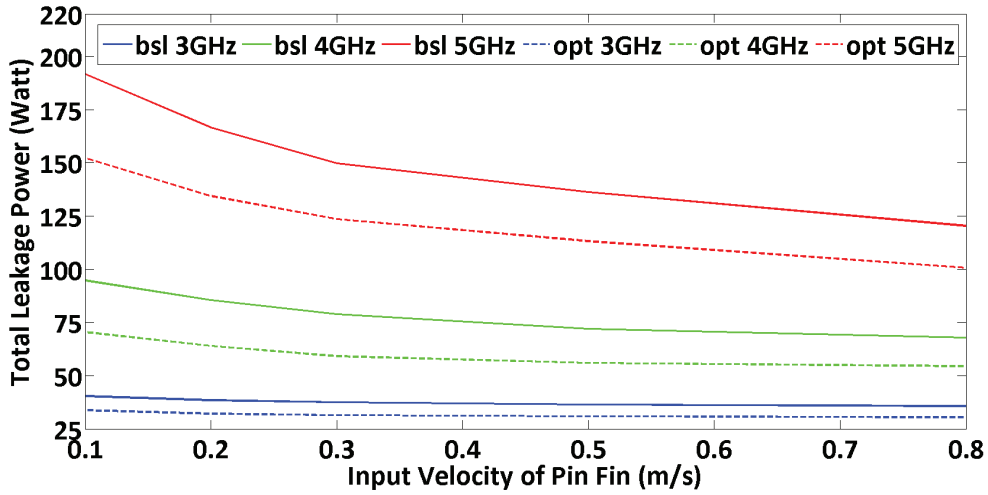
(b)

Figure 4.6: Power characterization in terms of core frequency in (a) barnes, and (b) ocean-c [3]

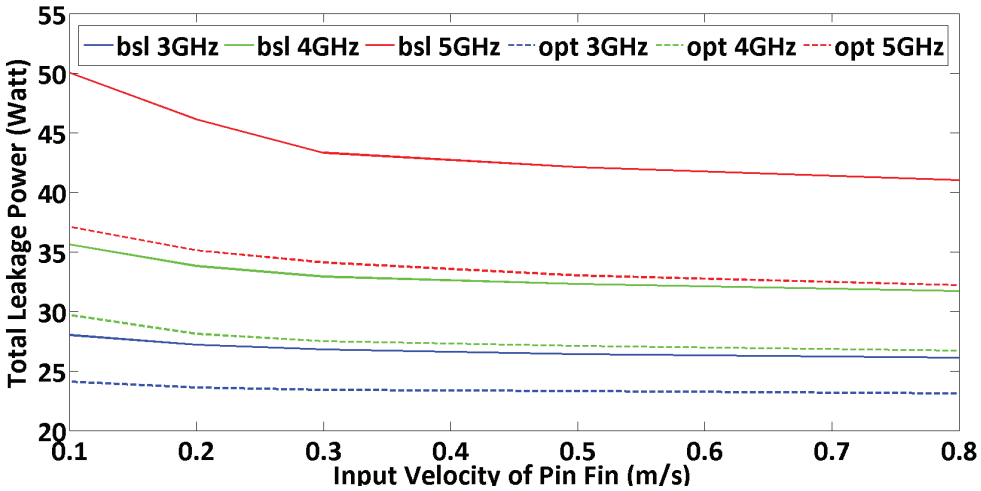
we characterize the total leakage power with respect to the fluid velocity of microfluidic cooling. Simulation results in Figure 4.7 indicate: i) the leakage power decreases with increased input velocity due to improved heat transfer capability, ii) system running at a faster clock frequency benefits more when increasing the fluid flow rate, as system at faster frequency tends to generate more power, iii) optimized pin fin configuration can provide significant improvement in heat transfer capability; for example, the leakage power of a $5GHz$ system is reduced by over 22% in all test cases at $0.1m/s$ Darcy velocity between a baseline and optimized configuration, iv) when the fluid flow velocity is small (i.e., below $0.4m/s$), the computational bounded application has more leakage power reduction than the memory bounded ones, as it tends to have a higher instructions per cycle (IPC) and generates more heat accordingly.

The leakage reduction in the optimized pin fin comes from the runtime thermal hotspot removal. Consider the exponential relationship between temperature and leakage currents [66]. At higher temperature (which means at higher frequency of operation) reductions in temperature will produce higher reductions in leakage current than at reductions in temperature that take place at lower temperature. Therefore, the leakage reduction in $barnes$ using the optimized pin fin is larger than in $ocean$-c, as $barnes$ executes at a higher temperature, as shown in Figure 4.7.

It is apparent that microfluidic cooling will enable the processor to execute at a higher frequency, compared to one with a conventional heat sink. Therefore, the system equipped with micro-pin fins can realize higher throughput. Figure 4.8 (a) compares the system throughput with respect to clock scaling. Because the memory system is on a separate (constant) clock, throughput gain in both applications do not increase in proportion to clock frequencies. The normalized system throughput of $barnes$ is higher than $ocean$-c since $barnes$ has a lower cache miss rate and thus fewer interactions with the slow system memory. Overall, the higher clock rates made feasible by microfluidics still enables overall 20% and 40% improvement in $4GHz$ and $5GHz$ respectively. This does not necessarily



(a)



(b)

Figure 4.7: Leakage power in terms of fluid velocity for two pin fin configurations in (a) barnes, and (b) ocean-c [3]

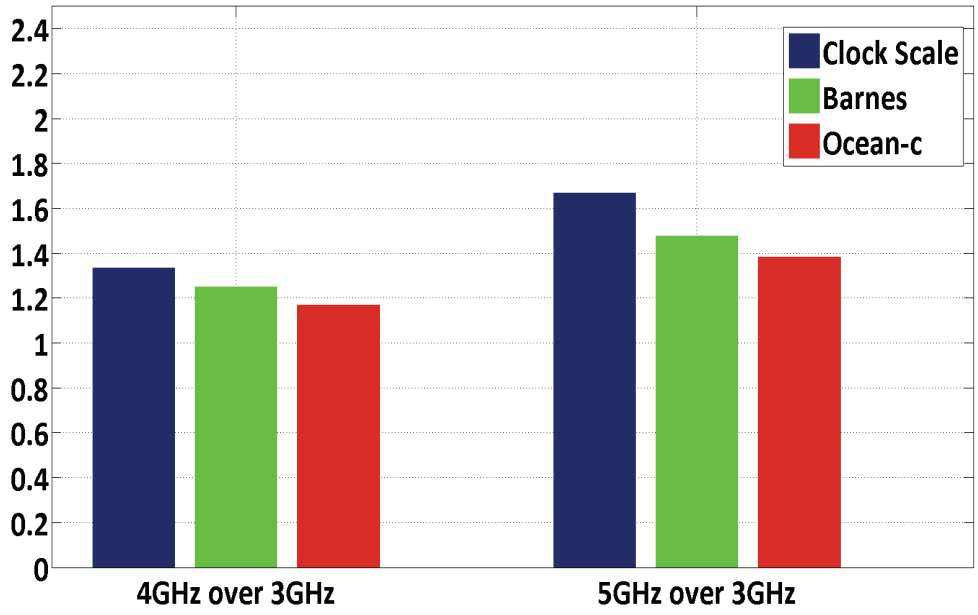
mean that the energy efficiency is improved as we describe later.

In addition, we evaluate the system energy efficiency in terms of energy per instruction (EPI), which tracks the average energy used to execute a single instruction. Figure 4.8 (b) illustrates the energy efficiency of the system with different frequencies under the optimized pin fin configuration as listed in Table 4.2. The input fluid velocity is set to $0.8m/s$. The EPI of $barnes$ from $3GHz$ to $5GHz$ keeps increasing, as $barnes$ operates at a relatively high temperature (above $350°K$) due to high IPC. The power dissipation grows faster than the reduction in execution time because of the quadratic relationship between leakage power and temperature. In contrast, $ocean$-c works around $330°K$, and the increase of leakage is approximately linear. The EPI of $ocean$-c remains approximately as a constant, because the system speedup from clock scaling compensates for the increase in system leakage power.

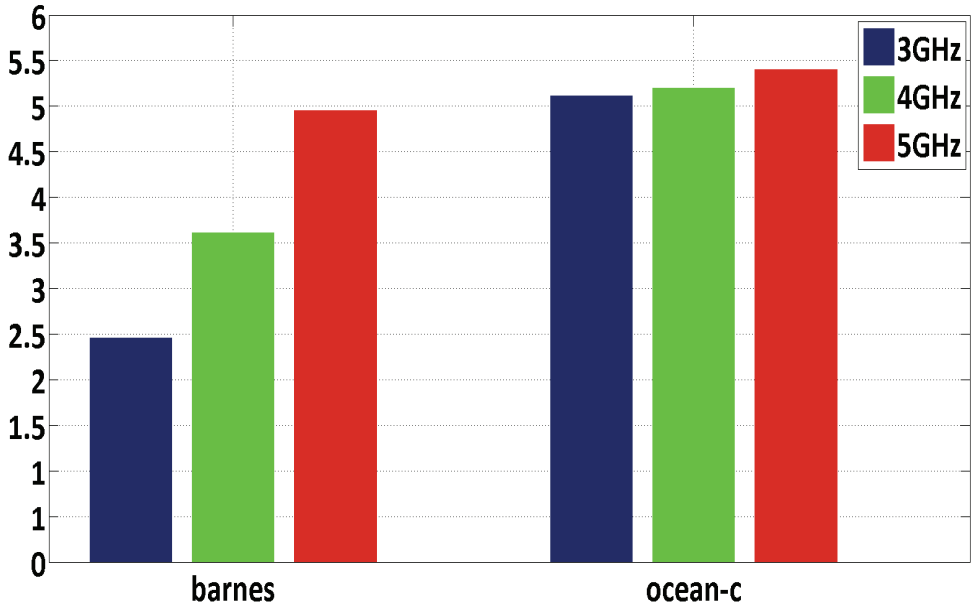
4.3 Thermally Adaptive Cache in 3D Many-Core Processors

4.3.1 Motivation

As CMOS technology advances, we are observing a confluence of technology and application trends in which the cost, execution time, and energy of applications are being dominated by the memory system. This is driving the industry to 2.5D and 3D packages for processor and memory systems. However, these packages also lead to higher heat fluxes and increased thermal coupling between the die challenging thermal solutions as demonstrated previously. When the core and cache are stacked, the temperature of the cache will be affected by the power activity of the cores as a result of the small inter-tier distance. In this situation, we cannot apply similar algorithms of computational sprinting from 2D processors to the 3D counterparts, since the increasing power consumption of the cores will bring up the cache temperature, which eventually leads to performance degradation of the entire system as a result of the much slower cache. Apart from our previous work that co-optimizes the cooling capability of micro-pin fins with power maps, we argue a



(a)



(b)

Figure 4.8: (a) System throughput, and (b) system EPI with respect to core frequency [3]

thermal-aware approach in processor design is a key to achieving high energy efficiency in the 3D system.

The key issue in 3D systems is that conventional design approaches utilize design margins that correspond to worst case temperatures and process corners. While such physical conditions may not occur often, the use of worst case design margins has a significant impact on average and peak system-level performance. We advocate for microarchitecture operational principles based on adaptation to thermal effects to improve performance over that achievable with designs based on worst case margins and demonstrate that this approach has considerable promise. A thermally adaptive mechanism is presented using a multi-physics modeling methodology, interacting with the available thermal headroom and circuit critical path delay during nominal operations. This approach differs from past ones focusing on maintaining the processor temperature below a peak value. In contrast, our techniques extend the dynamic operating range (voltage and temperature) of the processor and view the available thermal headroom also as a resource to be consumed for performance.

The target system is the same 16-core x86 homogeneous processor in a 3D stack organization as shown in Figure A.1. Microarchitecture parameters are given in Table 4.3. Applications running in the 3D processor exhibit a wide range of thermal behaviors affecting the temperature-dependent delay characteristics of the SRAM-based last-level-cache (LLC) producing temperature dependent access times. We provide a characterization of this delay behavior and propose two mechanisms for adapting to these delay variations. The first mechanism adapts the L1-LLC interface to vary the LLC access time as a function of temperature. The second mechanism adapts the core speed and scales the LLC frequency to match the time-varying LLC hit time. Using a full system simulator executing stock 32-bit x86 applications, we quantify the feasible performance gains and share some insights into the potential of this approach seeking to establish the need for, and value of, a multi-physics co-design approach for 3D microarchitectures for future processor design.

Table 4.3: Microarchitecture configuration parameters

Core configuration	
Fetch width	4
Execution width	5 (4 INT ports, 1 FP port)
InstQ size	32
ROB size	128
LSQ size	48 (32 loads, 16 stores)
Registers	
Cache configraution	
IL1	4-way 16KB, 1 cycle
DL1	8-way 32KB, 1 cycle
Last-level cache	32-way 2MB, 40 cycle

4.3.2 Thermal-Delay Characerization in SRAM Cache

The temperature-delay characterization in an SRAM bank is simulated with a $16nm$ HSPICE model depicted in Figure 4.9. The transistor sizing and cell configurations are optimized for the predictive model [67]. The critical path of a conventional SRAM bank is limited by the wordline driver, cell drive bit-line, sensamp sensing, and bit-line precharge/sensamp reset. This model assumes the wordline-reset is masked during sensamp evaluation with a divided-bitline multiplexing architecture. A latch-based sense amplifier architecture is considered for simulation of sense-amp delay [68]. Due to the regularity of the SRAM array, the extracted critical path of the sub-array is deterministic defined as:

$$T_{random-cycle} = T_{wordline-driver} + T_{cell-drive-bitline} + T_{sensamp} + T_{sensamp-precharge}. \tag{4.2}$$

According to Figure 4.9, the LLC bank access delay at $20°C$ is 54% of that at $85°C$. Figure 4.10 illustrates the IPC difference in systems with LLC runtime delay corresponding to $20°C$ and $85°C$. The baseline uses SRAM delay at $85°C$ as the worst-case design, while the ideal case keeps the SRAM delay corresponding to $20°C$. The IPC measurements are taken over $250M$ simulation cycles in the region of interest for each benchmark selected from SPLASH-2 [55] shared memory application suite. The geometric mean of the sys-

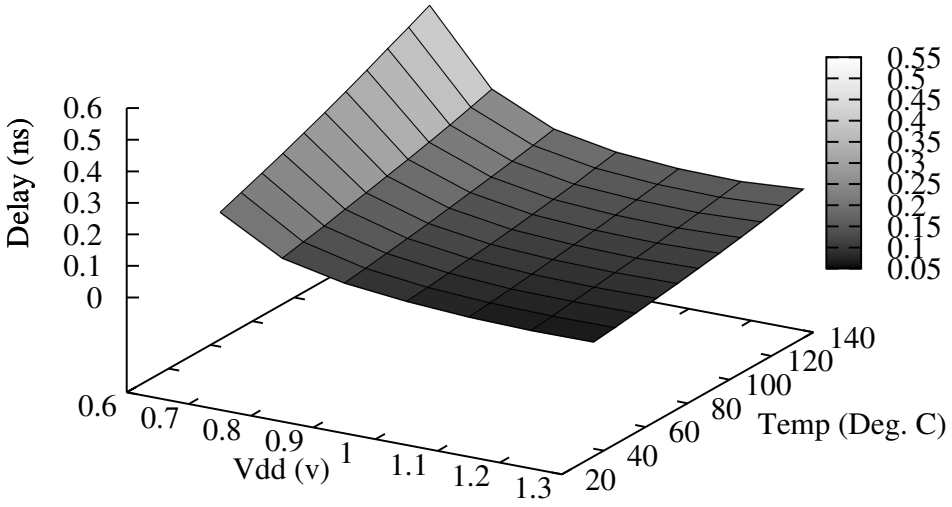


Figure 4.9: SRAM static delay model in terms of the supply voltage and temperature

tem IPC is improved by an average of 11%. $Barnes$ and $raytrace$ experience over 20% speed-up as they have a relatively lower L1 hit rate, but higher L2 hit rate. All the other applications achieve over 7% performance improvement except for $radix$ (2%). A closer look reveals that it is bounded by the memory latency, as it has the highest LLC miss rate. The results indicate the performance achievable with delay-dependent adaptation mechanisms.

4.3.3 Thermal Adaptation in 3D Processors

The basic idea of thermal adaptation we promote is to consistently convert thermal headroom into performance improvement. Specifically, we discuss two LLC adaptation models here in detail. The adaptation granularity is a critical factor for both models, as we need to make sure that the timing properties of SRAM do not change significantly within the sampling period. For a 2-tier 3D structure demonstrated in this thesis, a typical silicon

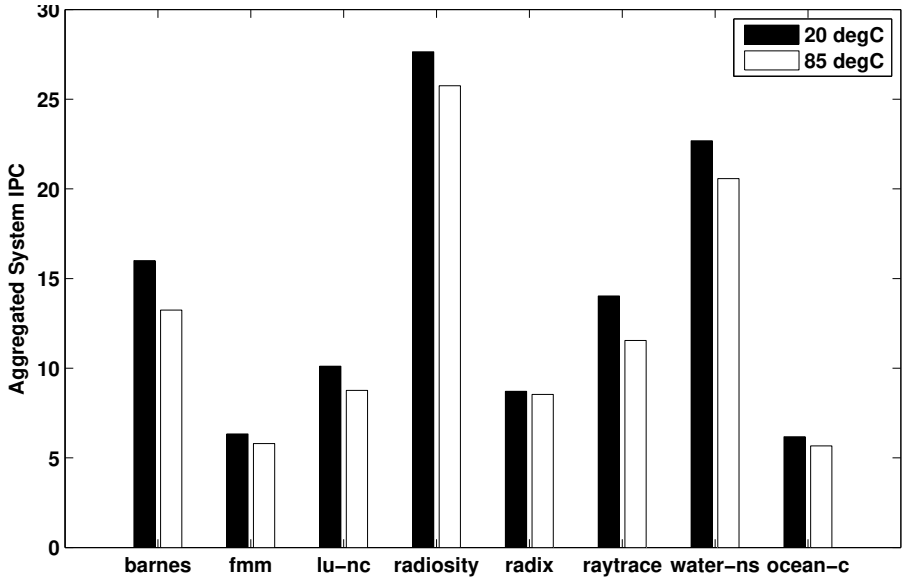


Figure 4.10: System IPC comparison between $20°C$ and $85°C$

thickness is $0.03cm$. The thermal time constant [69] is computed as:

$$\tau = (\frac{2 * th}{\pi})^2 \frac{\rho * C_p}{K} = (\frac{2 * 0.03}{\pi})^2 \frac{2.33 * 0.7}{1.0} = 609\mu s, \tag{4.3}$$

where ρ, C_p and K are the density, specific heat and thermal conductivity of the silicon. The temperature change in $10\mu s$ (the sampling period chosen in the simulation) for a single core with TDP of $20W$ is computed as:

$$\Delta T = P * R(\frac{4}{\pi^{1.5}})(\frac{t}{\tau})^{0.5} = 20 * 0.2 \frac{4}{\pi^{1.5}}(\frac{10}{609})^{0.5} = 0.37°C, \tag{4.4}$$

where P and R are the power consumption and thermal resistance of a single core. As shown, the temperature variation within $10\mu s$ is less than $0.5°C$. We propose the following two adaptive models.

4.3.4 Reduced Cycle Model (RCM)

RCM focuses on the interface between the core and its adjacent LLC cache bank. The RCM algorithm reduces the number of cycles to access the bank in proportion to the temperature

Algorithm 3 Thermal Adaptaion Framework

```
1: update_power(core[], cachebk[]);
2: update_temperatre();
3: synchronization_barrier();
4: for i = 0 to cache.banknum-1 do
   ▷ Reduced Cycle Model
5:     cachebk[i].cycle = cycle_tbl(cachebk[i].temp);

   ▷ Partial Boosting Model
6:     new_freq = core_boost(core[i].ipc,
7:             cachebk[i].temp);
8:     if power_avail(new_freq) > 0 then
9:         core[i].freq = new_freq;
10:    end if
11: end for
12: synchronization_barrier(); =0
```

drop during execution, and thus improves the cache performance.

As the temperature of the cache banks does not have significant changes within the sampling period, the new bank access time in number of cycles is updated as a function of the temperature at the end of the sampling period by indexing from a pre-computed lookup table of cache access in cycles. Support for the RCM is at the cache interface and does not affect the core hardware.

The performance gain in RCM comes from the reduced miss penalty in the L1 cache. RCM is suitable for memory bounded applications, as the applications have more cache interactions.

4.3.5 Partial Boosting Model (PBM)

Unlike RCM, PBM scales up the core frequency according to the temperature of its adjacent cache bank and the power budget, and tries to boost the frequency (and therefore voltage) of a core when the vertically adjacent LLC bank temperature is low. The voltage of LLC does not change during the period to keep a constant access based on the SRAM temperature-delay curve. Compared to conventional sprinting techniques, PBM uses the LLC bank

Table 4.4: SPLASH-2 benchmark characterization on a 16-core processor

App	uops	flops	mem read	mem write	L1 hit rate	LLC miss rate
barnes	2437M	11.9%	20.3%	15.6%	96.86%	16.97%
fmm	2624M	33.9%	18.1%	3.1%	98.13%	40.57%
lu-nc	415.9M	18.7%	21.1%	9.7%	93.55%	43.17%
radiosity	2891M	-	17.6%	10%	99.17%	17.36%
radix	325.8M	-	23.7%	13.8%	97.40%	44.65%
raytrace	719.6M	-	25.2%	9.6%	96.48%	24.70%
water-ns	675.1M	21.3%	17.6%	7.7%	98.62%	25.25%
ocean-c	665.4M	26.7%	21.6%	4.9%	93.55%	44.28%

temperature as a negative feedback to prevent system degradation from overheating.

At first, the core frequency is pre-set with respect to the IPC and temperature of its associated cache bank. We construct a analytical model of the upper bound on power in core and cache as a function of frequency and IPC. If the power budget (TDP minus estimated power at new frequency) is greater than zero, the core frequency will change to the new value. The maximum frequency is set to $4.5GHz$ to prevent system failure.

As PBM improves the performance of cores, computational bounded applications will realize greater performance gain.

4.3.6 Results and Analysis

To evaluate the effectiveness of thermal adaptation in 3D processors, we create a baseline 16-core homogeneous processor with no adaptive mechanism and compare against RCM and PBM. We characterize eight applications using the baseline configuration as shown in Table 4.4. The hit rate of L1 cache and the miss rate of the last-level cache are the geometric means of all 16 cache banks.

We first compare the performance of RCM and PBM in terms of IPC and MIPS. Figure 4.11 presents the IPC results. RCM has the best IPC, as its cache performance is improved. However, the IPC of PBM is worse than the baseline, as the cache miss penalty is increased as measured in *number* of clock cycle when the core boosts up.

Both RCM and PBM improve the system throughput shown in Figure 4.12. RCM

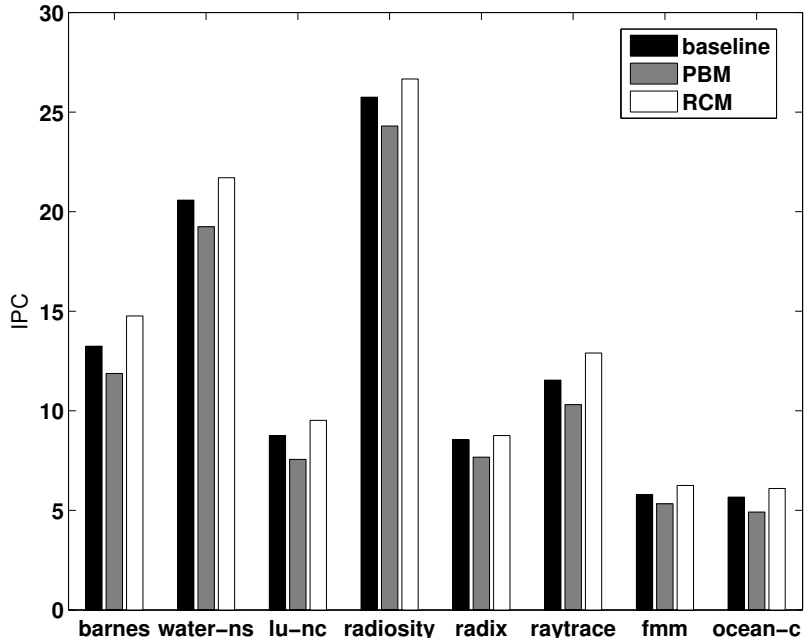


Figure 4.11: Comparison of RCM and PBM in instructions per cycle (IPC) [2]

speeds up the cache system by reducing the access time while PBM gains better throughput by boosting up the system clock. For typical computational bounded applications such as $radiosity$, PBM outperforms RCM by around 9% as more instructions can be executed from a faster core, yet for memory bounded application such as lu-nc, the performance of RCM is better than PBM by 5.2%.

For applications falling in between the computational and memory bounded categories, the situation might not be intuitive. The system throughput of $barnes$ is higher than that of $radix$, yet RCM outperforms PBM in $barnes$ by contrast. The reason is that the L1 hit rate of $radix$ is higher than $barnes$, so $barnes$ benefits more from improving cache performance and $radix$ gains more benefit from clock boosting.

Although system performance improves in both RCM and PBM, system power in these two systems incresases as well. The power consumption of the adaptive system is proportional to the system performance as shown in Figure 4.13, where $radiosity$ has the highest runtime power and fmm has the lowest value. For $barnes$, lu-nc and $raytrace$, the RCM power is higher than PBM as the system performance outperforms the PBM model. Gen-

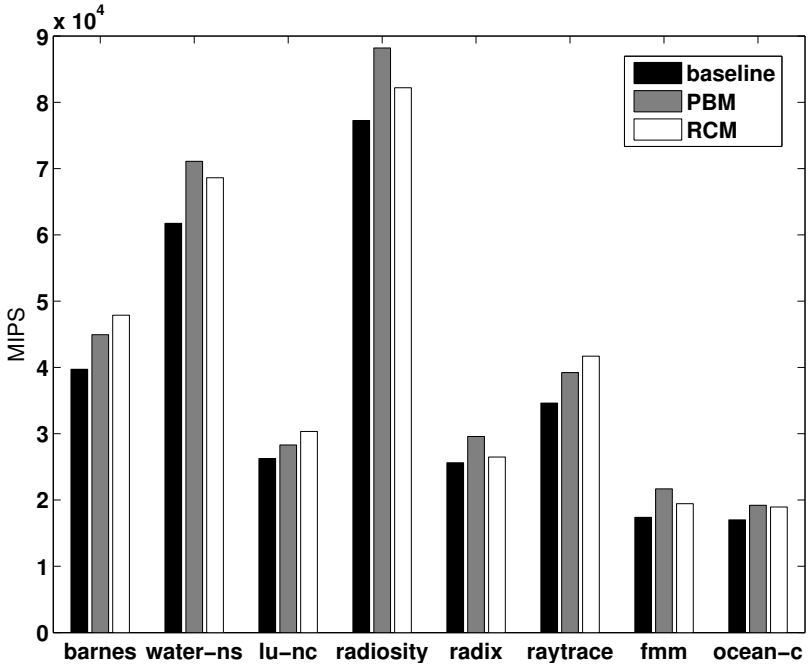


Figure 4.12: Comparison of RCM and PBM on system throughput (MIPS) [2]

erally, the PBM system consumes more power than RCM as both the core and cache run faster as shown in other five applications.

The total energy consumption of RCM and PBM reduces albeit an increased average power as shown in Figure 4.14. The dynamic power remains constant as application workload remains the same. However, the leakage energy decreases significantly, as the total execution time shrinks, as depicted in Figure 4.15. The only exception is $radix$. $Radix$ has the highest LLC miss rate (44.65%) and its performance is constrained by the memory system. As a result, the performance improvement brought by RCM and PBM does not compensate for the power increase. The total energy is thus increased.

Finally, we look at system energy efficiency measured by energy per instruction (EPI). EPI indicates average energy for a single instruction, as shown in Figure 4.16. For typical computational bounded applications such as $radiosity$, PBM achieves the best EPI, while the memory bounded application lu-nc gets the best EPI when applying RCM.

The LLC miss rate of $barnes$ is as small as 17%, but the energy efficiency of RCM

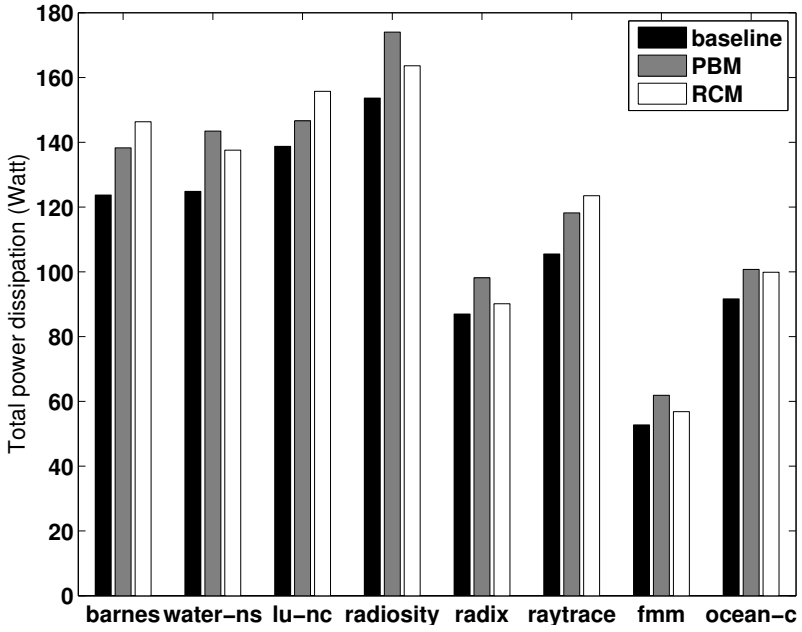


Figure 4.13: Comparison of RCM and PBM on power consumption (Watt) [2]

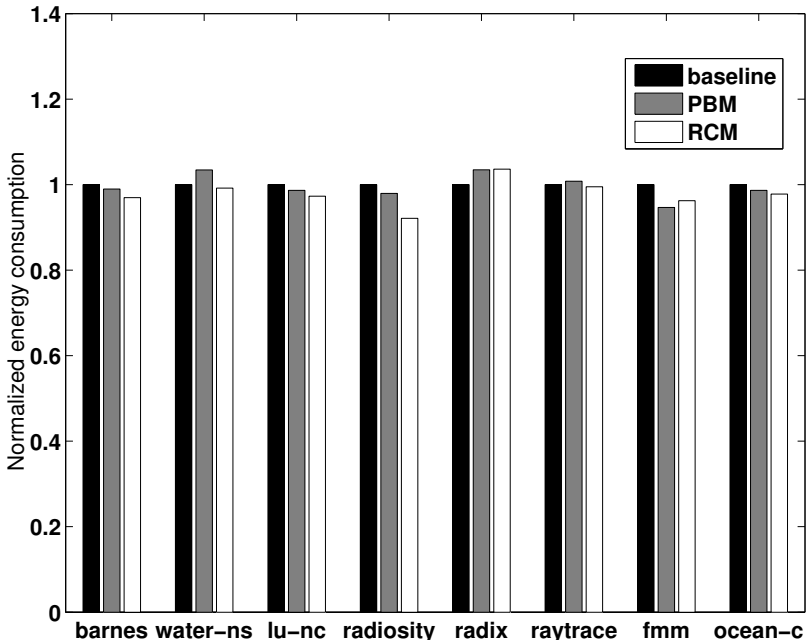


Figure 4.14: Comparison of RCM and PBM on normalized total energy [2]

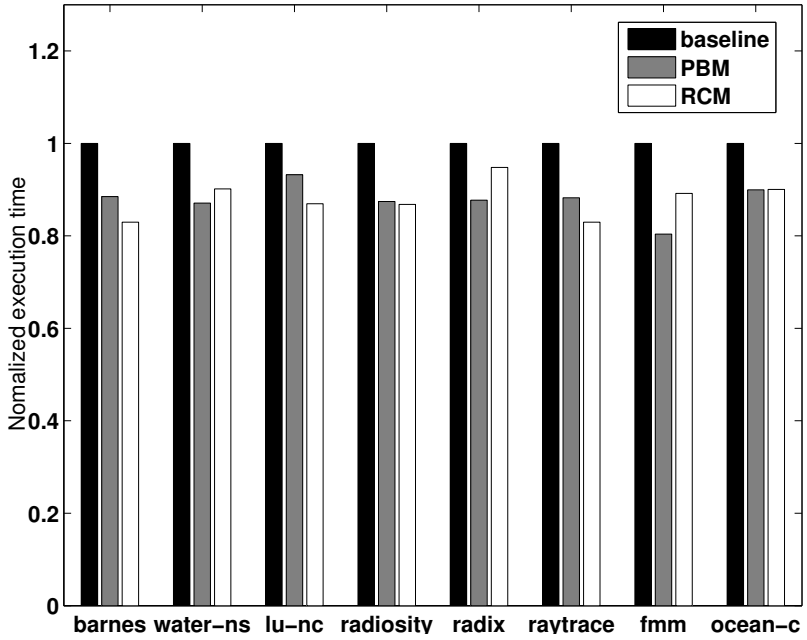


Figure 4.15: Comparison of RCM and PBM on normalized execution time [2]

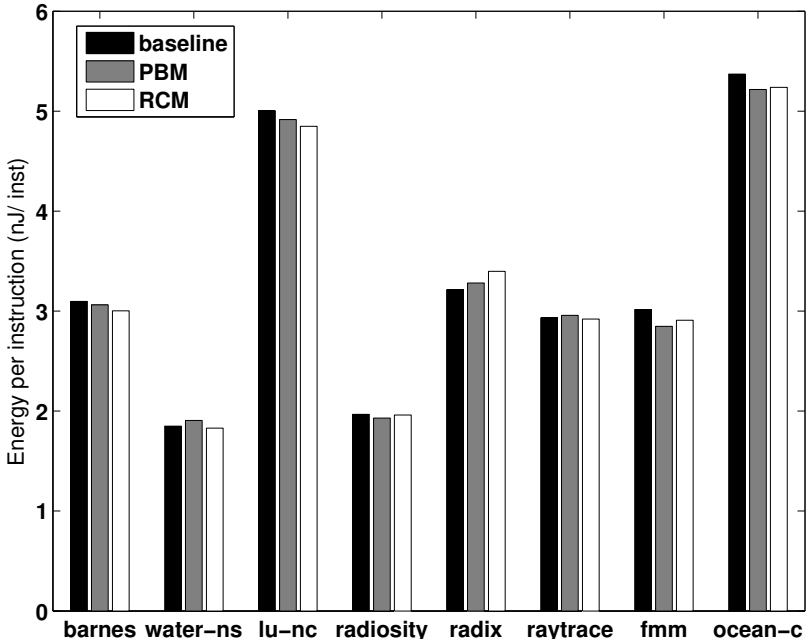


Figure 4.16: Comparison of RCM and PBM on energy efficiency (EPI) [2]

outperforms PBM, since the L1 hit rate of barnes is relatively small. $Barnes$ is thus sensitive to the LLC cache delay. As RCM provides the best LLC performance among the three models, it also provides the best energy efficiency. Moreover, the front-end temperature of the cores are high when running $barnes$, preventing PBM from running at a faster speed. On the contrary, fmm has LLC miss rate of 40.6%, yet PBM outperforms RCM in this application. This is because fmm contains a large amount of floating point operations, and PBM provides more benefit by boosting up the cores.

The only exception in our experiments is the $radix$ application. $Radix$ is bounded by the memory latency instead of the cache latency, as it suffers from high miss rate of the last level cache and contains no float point operations. For this reason, both adaptive models will not gain much benefit in performance, and the increase in power consumption in both cases will lead to energy inefficiency.

4.4 Summary

In this chapter, we characterize the thermal challenges in 3D processors because of its high heat flux and strong thermal coupling effect and demonstrate the feasibility of designing an high-performance 3D processor in a given thermal constraints through a co-design between processor microarchitecture and thermal cooling.

In the first part of our thermal co-design practice, we optimize the microfluidic pin fin using a holistic optimization framework based on 3D processor's floorplan and runtime power to minimize the thermal resistance within a fixed pumping power. Our results establish that an optimized pin fin structure with appropriate coolant velocity will enable the system to operate with a higher throughput and improved energy efficiency.

To the best of our knowledge, this is the first model and analysis that integrates into a single simulation model i) application binaries, ii) operating system binaries, iii) cycle-level multicore architecture timing, iv) power and energy models and v) thermal models. The self-contained simulation framework enables us to explore the impact of microfluidics

on computing system level metrics experienced by the applications and evaluate microarchitecture level metrics such as energy per instruction over various physical configurations.

In the second part of our thermal co-design practice, we argue for the multi-physics of processors as a driver for the design of 3D processors presenting a use case of a thermally adaptive LLC. Unlike previous efforts, the goal here is to consistently utilize all of the thermal headroom across the chip. Thermal headroom is a resource to be mined for performance and not a constraint to be met. We presented two thermally adaptive models for the LLC cache in a 3D stacking environment, RCM and PBM, to improve the system performance compared to conventional worst-case design operation. The RCM adapts the access time (in cycles) of the LLC cache to the temperature, while PBM modifies the core frequency based on the temperature of the vertically adjacent cache bank. Both models improves the overall system performance by over 20% and energy efficiency by up to 3%.

The thermal adaptation model we develop utilizes the circuit simulator to estimate a realistic temperature-delay model to trade off between thermal headroom and performance gain. We foresee understanding theses effects across new device technologies (e.g., FinFET vs. Planar, or eDRAM vs. SRAM). As the physical phenomenon increasingly manifests itself at the system level, this visibility across thermal modeling, circuit behaviors and microarchitecture design will become increasingly critical to fine-grained optimizations.

CHAPTER 5
CO-DESIGN OF PROCESSOR ARCHITECTURE AND POWER SUPPLY SYSTEM

5.1 Introduction

3D ICs suffer from unsustainable growth in power consumption and thus harm the power efficiency of processors. New advances are central to the effective operation of all modern processors in platforms ranging from mobile devices to data centers and high-performance computing (HPC) machines that drive national initiatives in key areas such as science, finance, and defense. Consequently, multiple efforts at various levels of abstraction have been developed for the power efficient design and management of multicore processors.

The most direct way to minimize power consumption and improve system power efficiency is through voltage reduction because of the quadratic relationship between supply voltage V_{dd} and power consumption. The dynamic power due to switching capacitances is proportional to $\alpha f C V_{dd}^2$; the static power is the power consumed from unintended current leakage at all junctions of VLSI devices and is exponentially dependent on V_{dd}.

A common practice for processor architects is to statically design the guardband of supply voltage for worst-case scenarios as shown in Figure 5.1. Therefore, the processor is running at higher voltage levels during nominal operations, which is a heavy burden for 3D processors with limited capability of power delivery and thermal dissipation. Granted that we relax guardband constraints in real-time, we can effectively minimize the runtime power consumption.

In this chapter, we advocate an adaptive design paradigm for power efficient processors through a co-design of circuit models and processor architecture. We demonstrate the necessity of such co-design paradigm with our work on adaptive designs both in the last

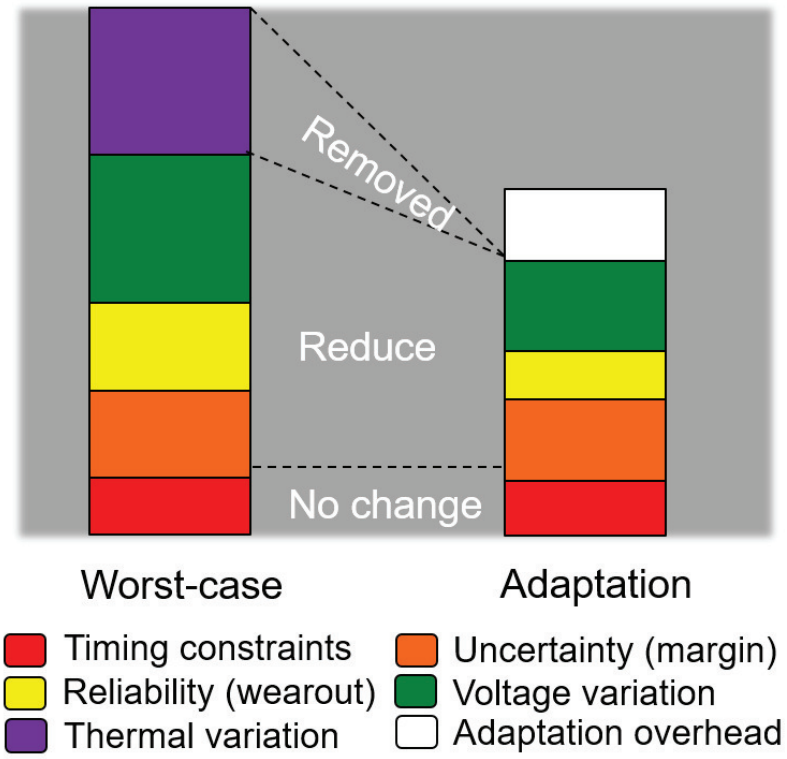


Figure 5.1: Voltage guardband breakups and possible optimizations [70]

level cache (LLC) and on-chip voltage regulator that achieve considerable power benefits in 3D multi-core processors.

5.2 Power Efficient LLC in 3D Processors Through Temperature Effect Management of SRAM Supply Voltage

5.2.1 Motivation

As described in Chapter 4, processor designers encounter challenges in heat dissipation within a 3D package. As operations of electrical circuits have a strong dependency on the temperature and the thermal coupling between layers exacerbates this dependence [61], the thermal challenges of 3D ICs expose significant performance penalties for such an approach since worst case conditions may not occur often on practice. Thermal behaviors are in fact driven by applications whose behaviors are time-varying.

In this section, we advocate an approach to converting the thermal headroom made

available from worst case design to improve energy efficient operations. This requires understanding and controlling coupled interactions between workload behaviors, microarchitecture power management, circuit adaptation techniques, and choices of packaging. Based on our characterization of the temperature-delay dependency of SRAM cells in 16nm technology, we understand the voltage margin available at each temperature relative to worst case design. Employing worst case design margins will fix the SRAM access delay corresponding to the worst case temperature. While maintaining this worst case delay, at lower temperatures we can lower voltage to maintain the performance (delay) but reduce energy consumption and thereby improve energy efficiency.

The state of the practice is to maximize performance for a given thermal budget in 3D ICs compared to convention approaches [71] [72], which emphasizes system performance over temperature considerations. In this section, we address the problem of maximizing energy efficiency for a given thermal budget. Our co-design approach is based on i) picking a system optimization objective (system level energy efficiency), ii) characterization of interdependencies (temperature-delay behavior), iii) understanding the consequential impact on applications (performance vs. energy efficiency), iv) devising online solutions for optimizing combinations of applications, architecture and circuits (temperature-driven dynamic adaptation of voltage margins), and v) assessing the gains for alternative packaging options (2.5D and 3D).

5.2.2 Impact of Package Configurations

We extend a 16-core 3D processor with a separate core and LLC layer to a complete system including a DRAM main memory, and construct two types of package configurations based on the placement of the DRAM memory, as shown in Figure 5.2. In the 2.5D package configuration, DRAM shares a silicon interposer with the 3D processor. We model a face-to-back interconnection between cores and the cache with a BEOL metal layer and communications between processor and main memory are through the wires that reside in

Table 5.1: Memory parameters in 2.5D and 3D package configuration

Memory configraution	
2.5D Memory	4 channels, 50ns per access
3D Memory	16 channels, 30ns per access

the silicon interposer. In the 3D package configuration, the main DRAM is placed on top of the processor structure as a stacked DRAM similar to a Hybrid Memory Cube [73]. The interconnection between LLC and DRAM here is configured in a 2D torus topology through TSVs. We assume a conventional forced-air cooling on top of both packages, which attach directly to a copper heatsink with a dimension of $50mm \times 50mm \times 20mm$.

The memory bandwidth in the 3D package is 4 times of that in the 2.5D package, and the latency of each memory request in the 3D package is improved by 30% compared to a 2D package [74]. DRAM parameters in 2.5D and 3D packages are Table 5.1.

The system performance and power profile depends on the DRAM memory configuration as well. When the DRAM memory is stacked on the LLC cache tier (pure 3D package), the overall system has a much larger memory bandwidth and a lower memory request latency. As a result, the cores tend to consume more power and generate more heat, and the thermal coupling between cores and the LLC is significant. Moreover, the stacked DRAM adds thermal resistance to the heat sink, and the cooling capacity is reduced. Therefore, the power/energy improvement from the constant performance model (CPM) scheme is limited compared to the use of 2.5D packaging where the DRAM is instead placed on the silicon interposer.

We compare the system performance between the 2.5D and 3D packaging in terms of instructions per cycle (IPC) running eight applications from SPLASH-2, as shown in Figure 5.3. The memory bound applications are more sensitive to the packaging differences because of the intensive interaction between processors and DRAM memory. For example, the *lu-nc* application suffers from over 50% performance degradation from 3D to 2.5D,

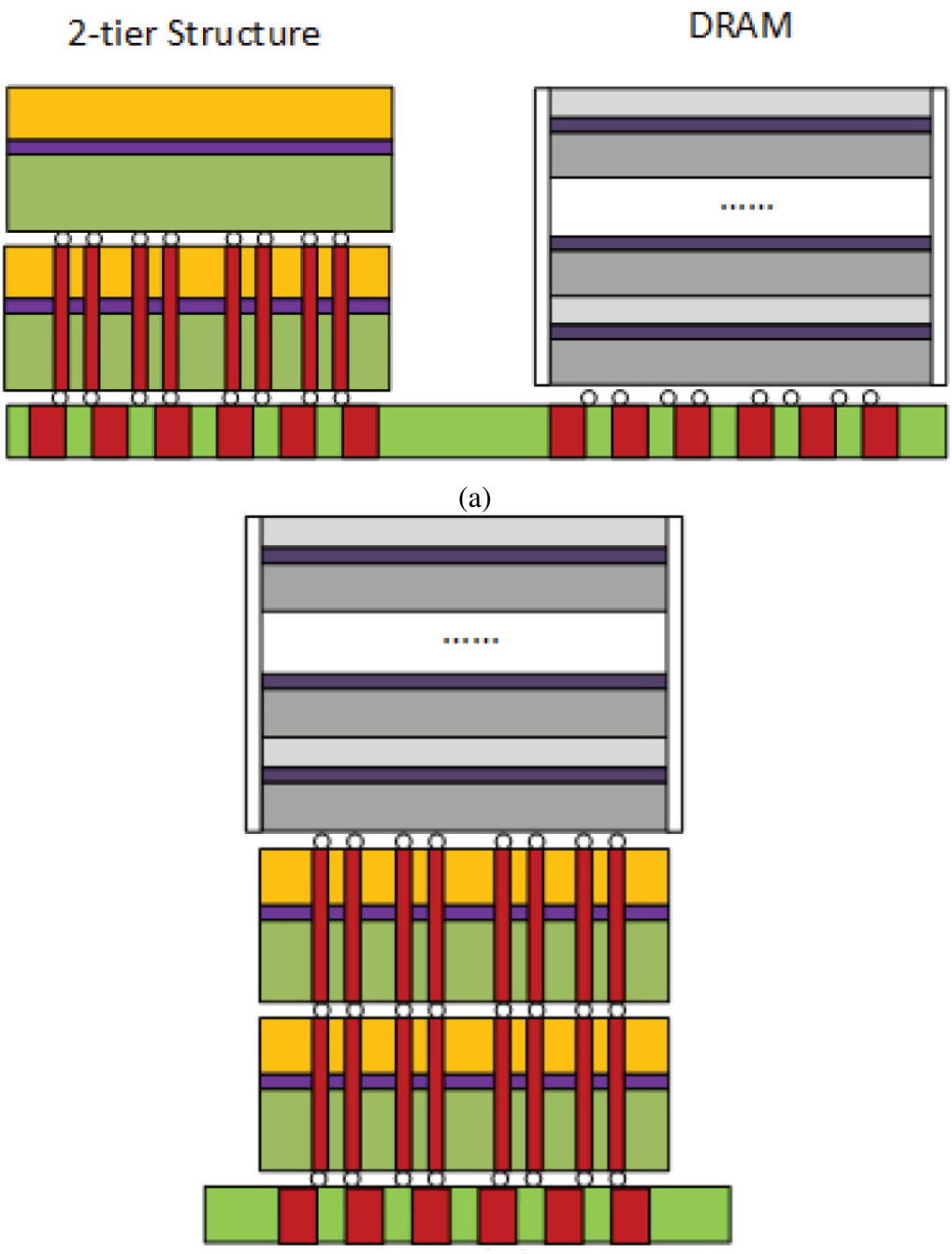


Figure 5.2: Package configuration of processor with DRAM in (a) 2.5D, and (b) 3D

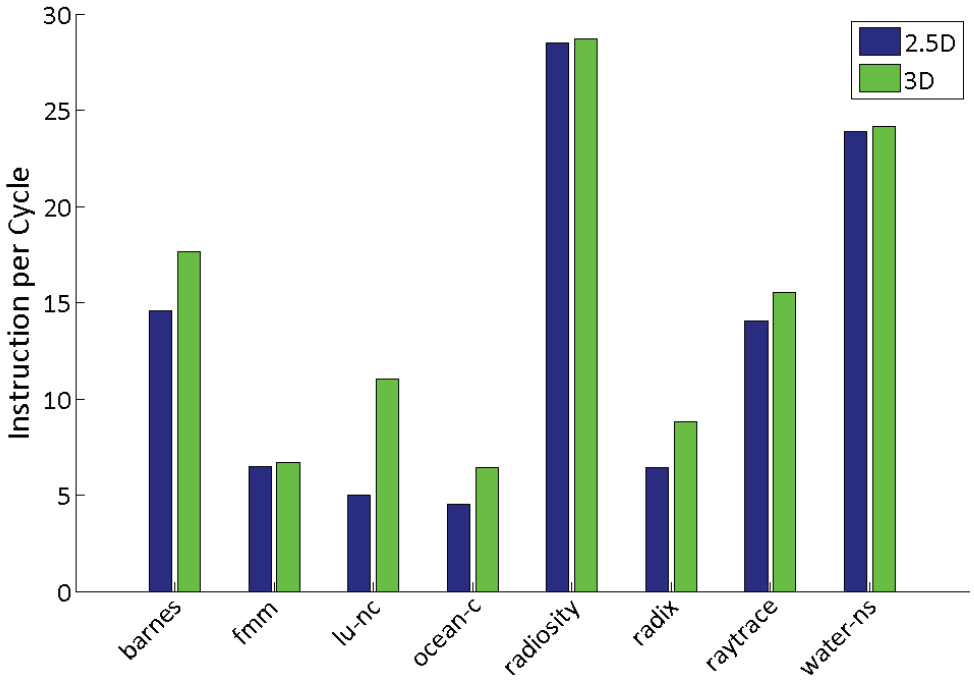


Figure 5.3: System performance comparison between a 2.5D and 3D package in terms of IPC

while the *water-ns* application has only 0.9% IPC reduction.

There are three main reasons for the impact of the 2.5D packaging on system performance: i) DRAM bandwidth is reduced, as the available channels in 2.5D packaging is limited compared to 3D, ii) the DRAM access time is higher, and iii) the average routing distance and therefore latency between the LLC cache and the DRAM controller is longer.

5.2.3 Constant Performance Cache

We utilize the SRAM HSPICE model described in Chapter 4 to explore the thermal dependency of the LLC in the 2-tier 3D processor. Specifically, we implement a SRAM bank model with thermal interaction and sythesize the sub-array of SRAM with a schematic-level memory compiler for a given configuration of memory arrays. Because of the regularity of SRAM, the extracted critical path of the sub-arry is deterministic, as shown in Figure 5.4, determined by the wordline driver, cell drive bit-line, sensamp sensing, and

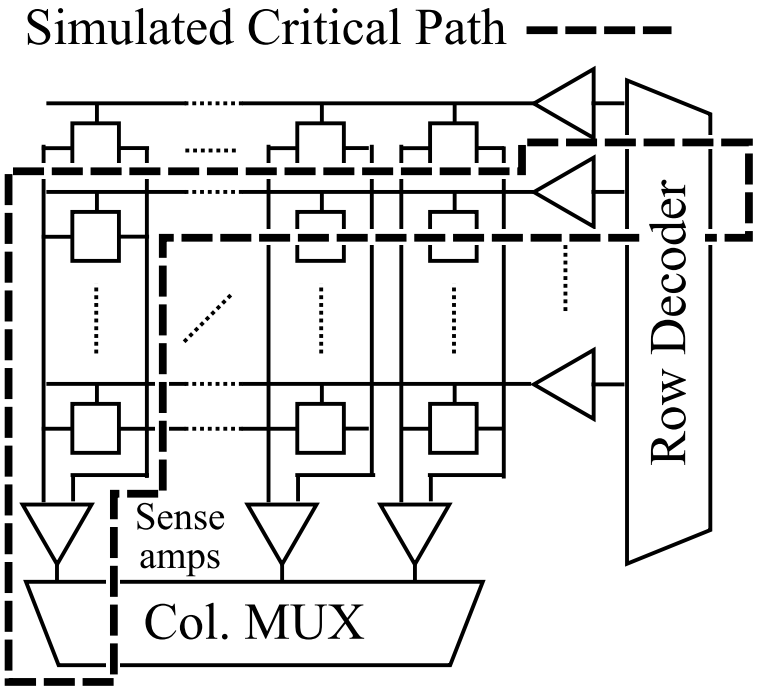


Figure 5.4: The delay model of a SRAM sub-array critical path

bit-line precharge/sensamp reset. This model assumes the wordline-reset is masked during sensamp evaluation with a divided-bitline multiplexing architecture.

Based on our observation, we propose a constant performance model (CPM), to fully utilize the thermal headroom during operation. Since the I_{on} current of a CMOS device is a quadratic function of the supply voltage and I_{off} current is an exponential function of V_{dd}, the supply voltage scaling is an effective way to reduce the power consumption. The CPM model, derived from dynamic voltage scaling, regulates the supply voltage of the SRAM cache banks individually to reduce the runtime power consumption. Initially, the voltage of each bank corresponds to the maximum SRAM access delay which corresponds to that for maximum temperature, that is, worst case conditions. The goal of the CPM is to enable bank-level voltage regulation in SRAM LLC cache, to dynamically reduce the unnecessary voltage slacks at lower temperature and to mitigate the effects of using worst-case design voltage margins. The voltage can be reduced without compromising timing integrity since the critical path delay also reduces. Therefore, CPM decreases the supply voltage of the

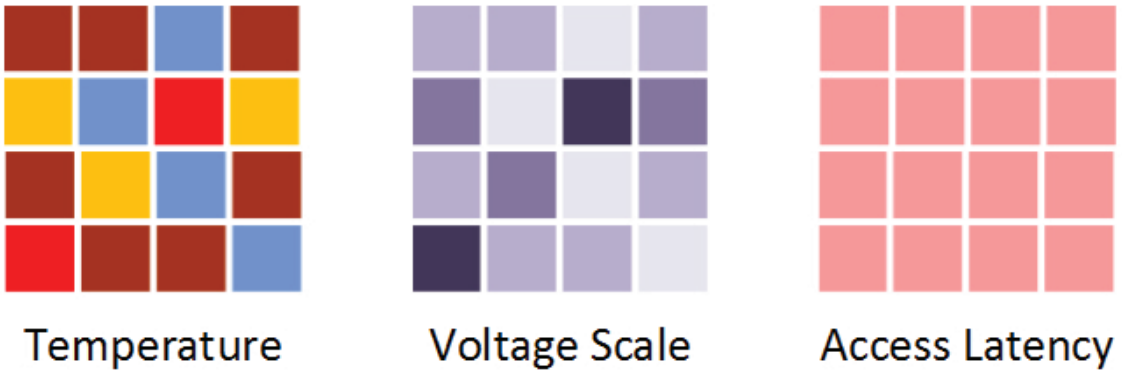


Figure 5.5: Runtime snapshot of CPM with temperature variation between cache banks

cache bank according to the temperature level of that bank while maintaining a constant cache access time, as depicted in Figure 5.5, based on the SRAM temperature-dependency curve. Since the cache latency remains constant throughout execution, system performance will not be degraded.

Meanwhile, as the voltage drop of the cache banks reduces power consumption, it provides a positive feedback to reduce the temperature of the whole system. Figure 5.6 demonstrates the improvement of the thermal behavior of the SRAM cache tier when running the typical memory bounded application *lu-nc* - one of the benchmark applications used in this analysis. The maximum temperature (hotspot) is reduced by around $8°$K.

5.2.4 Voltage Adaptation Algorithm

Our baseline LLC cache operates at $0.8V$ in $3GHz$, and the hit time is 30 cycles. Worst case thermal conditions lead to bank temperatures of approximately $400°K$. To characterize the temperature dependency of the cache, we run simulations across wide range of voltages - $0.6V$ to $1.1V$ and corresponding temperatures. Figure 5.7 illustrates the results of this analysis showing the voltages required to maintain this baseline SRAM latency (corresponding to $0.8V$ under $400°K$) at different temperatures. The supply voltage can be reduced to $0.66V$ without any performance degradation when a cache bank temperature

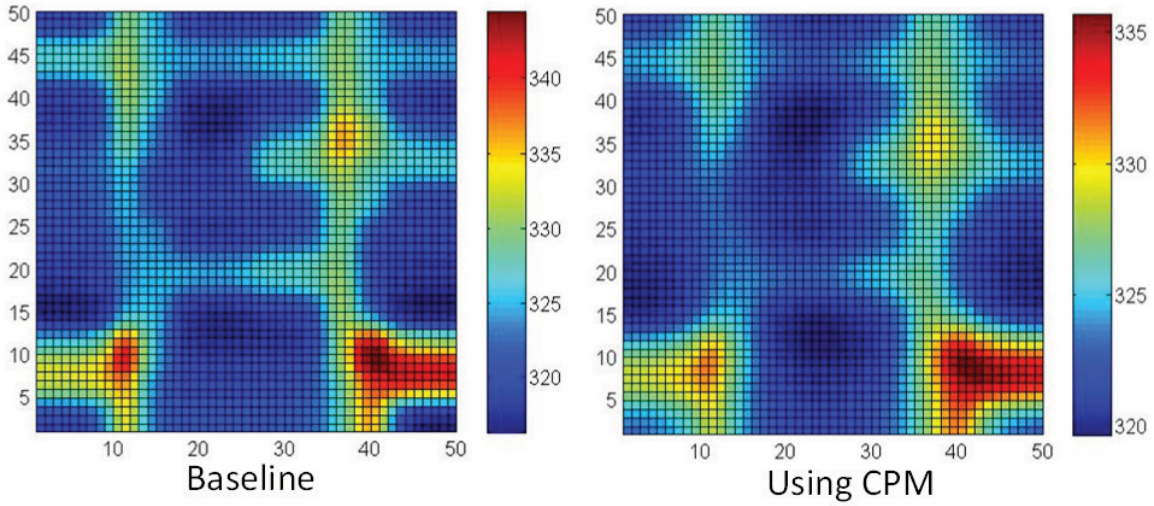


Figure 5.6: Temperature hotspot between baseline and CPM running the lu-nc application

drops to $300°K$.

When LLC initializes, supply voltage of all 16 banks sets to $0.8V$. The voltage is then scaled down when its temperature is below the scaling threshold. By assuming an ideal integrated voltage regulator (IVR) , the voltage changes complete instantly.

The basic idea of thermal adaptation is to trade off the circuit timing headroom with supply voltage reduction in the SRAM cache. The algorithm is depicted in Algorithm 4. The new voltage of the cache bank is determined by both the current temperature of the cache bank and the power of its associated core. The timing margin of the SRAM access can be directly calculated using the temperature of the local cache bank, and then be converted to the correct voltage drop. The power of the associated core gives hints as to the pipeline execution performance, and sets up the minimal voltage constraints for the SRAM cache bank to guarantee correct functionality. The new voltage is updated by striking a balance between the two parameters.

For compute bounded applications, the temperature of the LLC bank is largely affected by the activity of its associated core. As there are relatively fewer LLC access when running these type of applications, we only need to maintain the minimal supply voltage for a cache

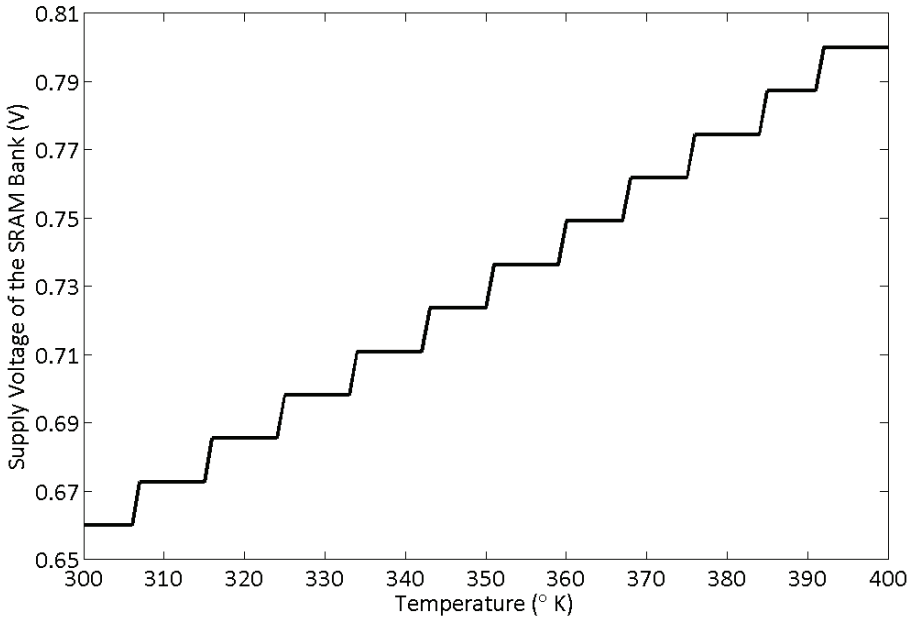


Figure 5.7: Supply voltage scaling of SRAM LLC to maintain constant access latency with respect to temperature

Algorithm 4 Thermal adaptaion in the SRAM supply voltage

```
1:  function AdptFrmwrk(void)
2:      updatePower(core[], cache[]);
3:      updateTemperatre();
4:      synchronizationBarrier();
5:      for i = 0 to cache.banknum-1 do
6:          cache[i].volt=updateVoltage(cache[i].temp,
7:                          core[i].power);
8:      end for
9:      synchronizationBarrier();
10: end function
11:
12: function updateVoltage(cacheT, coreP)
13:     index=genIndx(cacheT, coreP);
14:     newVolt=voltTbl[index];
15:     return newVolt;
16: end function
```

Table 5.2: Cache behavior characterization of SPLASH-2 benchmark

App	L1 hit rate	LLC miss rate 2.5D	LLC miss rate 3D
barnes	96.9%	16.4%	17.0%
fmm	98.1%	40.0%	40.6%
lu-nc	93.6%	45.3%	43.1%
ocean-c	93.6%	44.2%	44.3%
radiosity	99.2%	17.8%	17.4%
radix	97.3%	43.8%	44.7%
raytrace	96.5%	25.0%	24.7%
water-ns	98.6%	25.1%	25.3%

access. In contrast, the power consumption of the cache banks is much larger when the system executes memory bounded applications, and thus the bank temperature is mainly determined by the activity of the LLC bank. For the other applications, the voltage drop is the result of a combination of memory and compute behaviors.

5.2.5 Results and Analysis

We evaluate CPM relative to a baseline system with LLC supply voltage fixed to $0.8V$. The test applications are picked up from the SPLASH-2 benchmark, the cache characterization of which is depicted in Table 5.2.

The hit rate of the L1 cache and the miss rate of the LLC cache are calculated as the geometric mean of the cache banks. The compute bound applications have a high L1 hit rate, and most of the memory request are can be served in the L1 cache (e.g., *radiosity*). The memory bound applications otherwise have a high interaction with the LLC and the main memory (e.g., *lu-nc* and *ocean-c*).

Meanwhile, we construct an ideal system to capture the upper bound of power efficiency of the CPM model - this model maintains the delay of the SRAM corresponding to $300°$K and sets the supply voltage to $0.66V$.

The power reduction of the LLC cache using CPM is shown in Figure 5.8. The CPM reduces overall 15% maximum SRAM power among the eight applications, and an aver-

Table 5.3: LLC temperature hotspot between the baseline and CPM

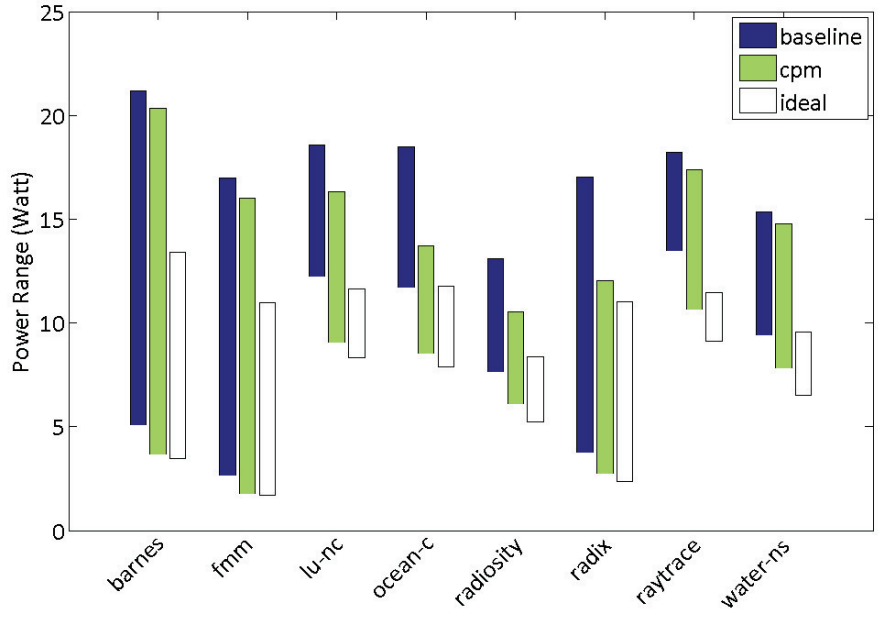
Δ T ($^\circ$K)	2.5D DRAM	3D DRAM
barnes	6.9	4.6
fmm	5.7	5.2
lu-nc	8.8	7.4
ocean-c	7.4	4.9
radiosity	2.6	2.5
radix	4.4	4.6
raytrace	5.1	3.2
water-ns	4.1	2.3

age of 23% of the minimum power both in the 2.5D and 3D packages. The significant power saving of the comes from the reduction of the unnecessary voltage margin. Memory intensive applications such as *ocean-c* saves up to 30% of the power.

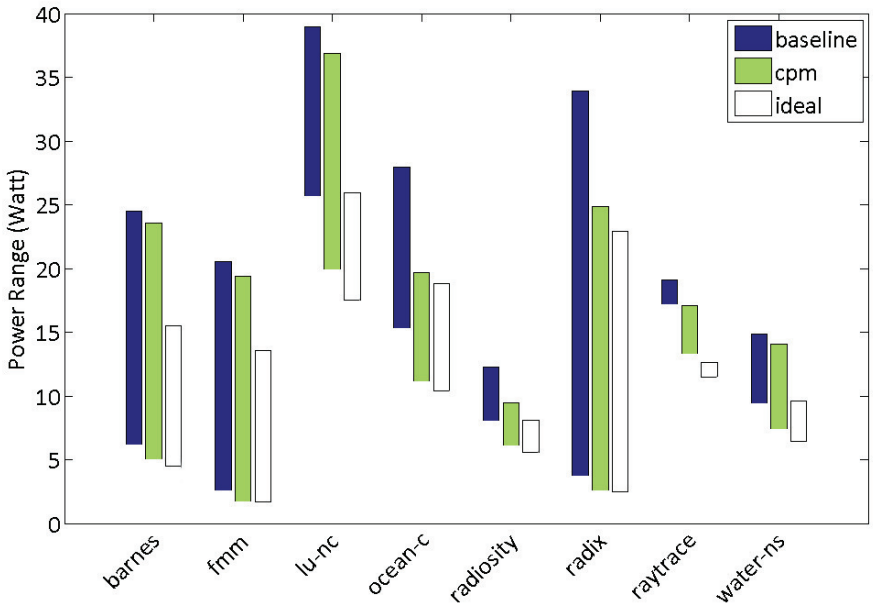
CPM also reduces the total energy consumed in the system throughout the execution. Figure 5.9 illustrates the normalized energy reduction of the LLC cache. The energy saving of the SRAM system is over 20% in average when CPM is deployed.

The voltage drop in LLC will also reduce the temperature of the cache bank, which in turn helps to further decrease the voltage of the LLC bank. As shown in Table 5.3, the CPM will reduce the hotspot of the SRAM cache by an average of 5°K. The 2.5D package has a little better temperature reduction, as the system runs slower than the system with 3D DRAM stacking. The temperature of the 2.5D system is lower, enabling greater voltage drop during execution. The *lu-nc* application has the largest temperature reduction of 8.8°K and 7.4°K respectively in the two package configurations for two reasons. First, it is a memory bound application, and the power reduction is significant when there is a voltage drop in the cache; second, it has the largest amount of LLC activity of all memory bound applications.

The power consumed in the LLC cache can take up 10% to 35% of the total power of the 3D structure, and the runtime SRAM power reduction will improve the system energy efficiency in terms of energy per instruction (EPI), as shown in Figure 5.10. The EPI

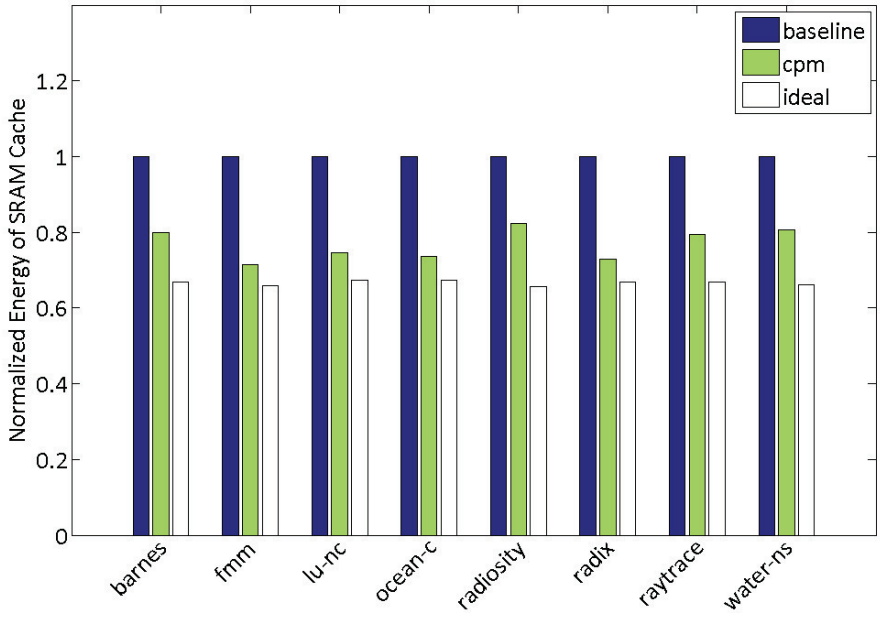


(a)

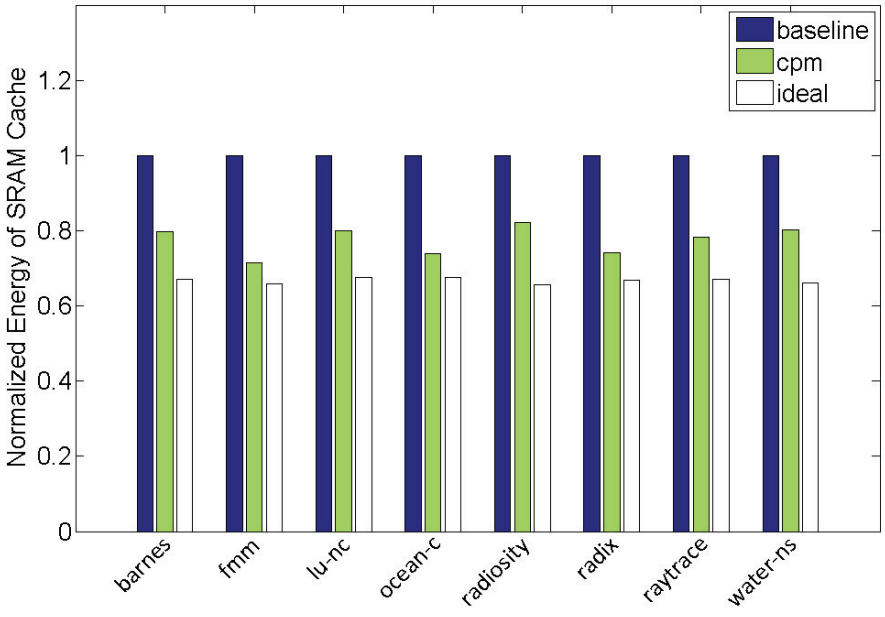


(b)

Figure 5.8: Runtime power profile between systems with: (a) 2.5D DRAM, and (b) 3D stacked DRAM [4]

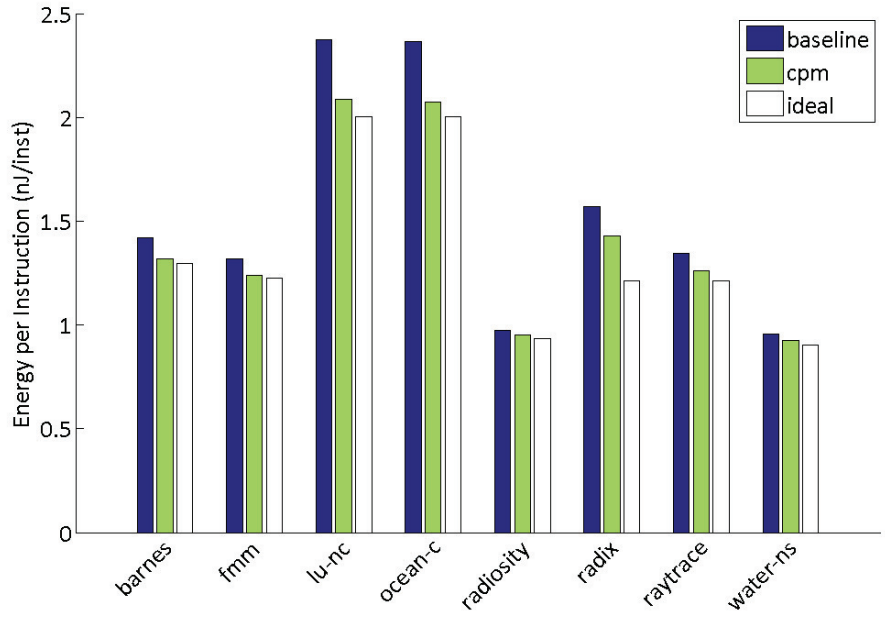


(a)

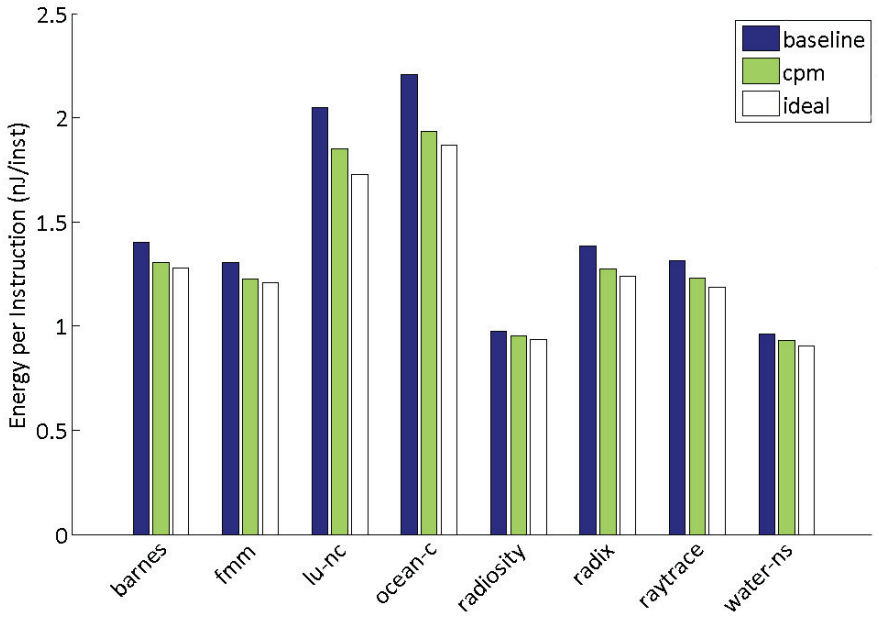


(b)

Figure 5.9: Normalized SRAM energy saving between systems with: (a) 2.5D DRAM, and (b) 3D stacked DRAM [4]



(a)



(b)

Figure 5.10: Energy efficiency between systems with: (a) 2.5D DRAM, and (b) 3D stacked DRAM [4]

records the average energy used to execute a single instruction. For both of the 2.5D and 3D configurations, the memory bound applications have better EPI improvement (11%) than compute bound applications (5%), as the proportion of power consumption in the cache system is higher.

5.3 Voltage Variation Prediction for Processor Transient Loads and Energy Efficient Power Management Design

5.3.1 Motivation

In the previous section, we propose an adaptive design for an SRAM LLC to reduce the thermal guardband in the supply voltage to achieve high energy efficiency during execution. In this section, we promote a self-adaptive mechanism integrated into on-chip voltage regulator of 3D processors to minimize transient variations and to reduce the required voltage guardband. A particular challenge has been to balance the needs of throughput related performance against the energy minimization needs of power efficient computing. In this section, we present a cross-layer technique for coordinating power delivery and power consumption to realize gains in power efficiency with no impact on perfomance. Power consumption behavior at the microarchitectural level is utilized in a predictive manner to modify the design of on-chip voltage regulator for more power efficient processor operation.

To prevent timing violations in critical paths experiencing a voltage droop, circuit designers deploy a voltage guardband in supply voltage based on worst-case voltage droop and thus processors operate at corresponding higher voltage. Such a conservative scheme of designing the power delivery systems exacts power and energy efficiency costs. These effects have a greater impact in 3D processors where the dark silicon effect is more pronounced. Consequently, to improve power efficiency there has been considerable effort on reducing voltage noise and supply [75] [76] [77] [78]. Several voltage droop management circuits uses reactive control to mitigate large droops through current sharing [79]. We

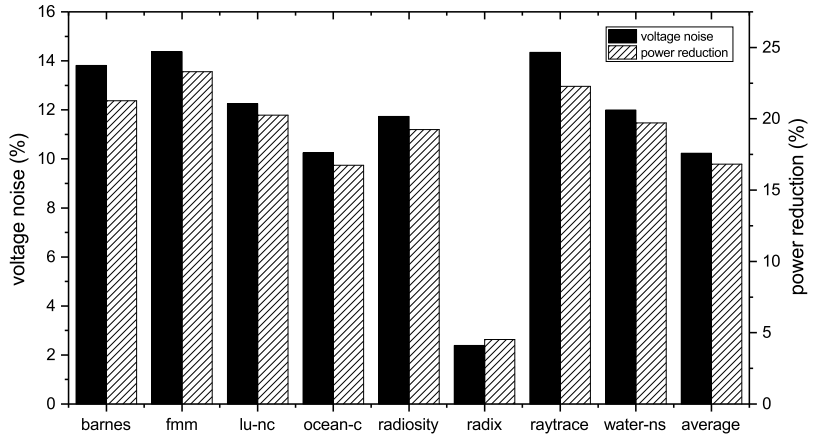


Figure 5.11: Comparison of voltage guardband and power reduction in a 4-core 3D processor executing the SPLASH-2 benchmark

investigate the opportunity of power reduction from voltage smoothing provided an ideal voltage regulator, which outputs a constant $0.85V$ regardless of load transients. As shown by the dash bars in Figure 5.11, we can obtain a 16.8% power reduction on average for analyzed applications. To bridge the gap between existing techniques for voltage regulation and the ideal scenario, we promote a reliable prediction of load transients in voltage regulation, which reduces the guardband with consequent power/energy savings.

There has been considerable prior work on the design of voltage regulators in the broader context of efficient power delivery. Our efforts seek to complement these efforts based on the insight that microarchitectural events can serve as good predictors of impending increases in current demands. Power dissipation in high performance out-of-order cores has a complex relationship to microarchitectural events such as cache misses. We use off-the-shelf learning algorithms to construct models of these relationships that can subsequently be used on-line to predict upcoming current load variations. The predictive model is integrated with an on-chip voltage regulator that is designed to utilize this predictive information to reduce the voltage guardband. Therefore, the processor can operate at a lower supply voltage leading to reduced power operation and lower energy consumption.

As we note in Section 5.3.4, the evaluation focuses on the microarchitectural level prediction model. The IVR design we use is simplified, while practical IVR design that we

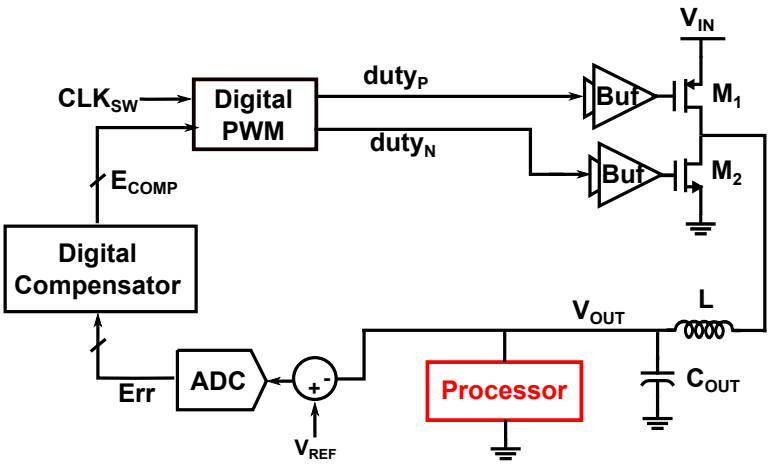


Figure 5.12: An on-chip voltage regulator model using a buck converter [81]

integrate the load prediction scheme needs to be more advanced.

5.3.2 Integrated Voltage Regulator and Voltage Droop

The operation of the power delivery system has a significant impact on the power/energy efficiency of processors. One key component is the voltage regulator, which delivers the power from an input source to processor circuits. Recently voltage regulators in modern processors have advanced to an on-chip implementation [80] in anticipation of its capability for fast voltage switching and fine-grained voltage control. We report on developments with an on-chip voltage regulator that is an inductor-based switching regulator whose design is shown in Figure 5.12. The switching circuit of the regulator that converts a DC voltage to a lower voltage with the same polarity by alternately connecting and disconnecting the source to an output inductor through pulse width modulation (PWM) control.

At load transients, the output of voltage regulators swing. When the IVR load current changes by Δi, the output voltage swing Δv is determined by the regulator capacitor C and transient time t, as given in Equation (5.1).

$$\Delta v = -Q/C = -\Delta i \times t/C = t/C \times -\Delta i \tag{5.1}$$

We apply step current with varied magnitudes at the IVR load and collect the output

voltage of the regulator. Figure 5.13 indicates a linear relationship between height of current steps and output voltage variations. To better understand the transient behavior of an on-chip voltage regulator, we apply multiple step signals of current draw with varied magnitudes that mimic load transients and collect the voltage variations at the output of the regulator. The simulation results are presented in Figure 5.13.

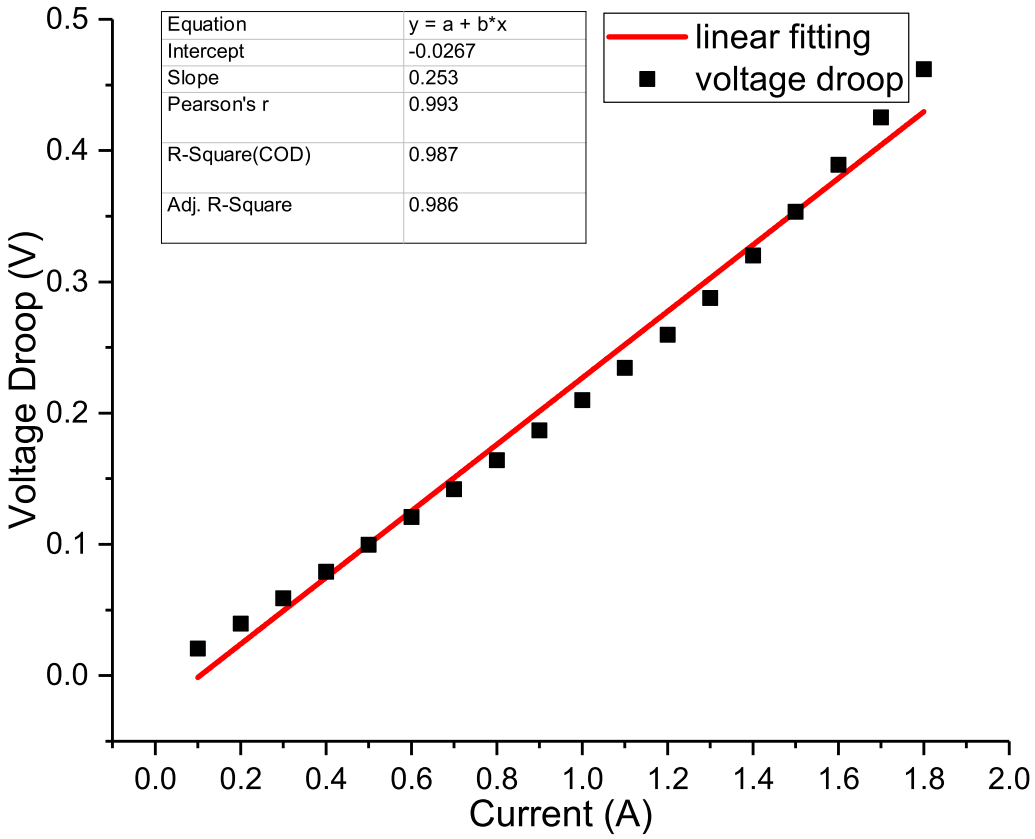


Figure 5.13: IVR I-V characterization

Figure 5.13 implies a linear relation between the input current and output voltage. When the input current through the inductor in Figure 5.12 changes by Δi, the voltage swing Δv equals to Δi times a constant determined by the regulator capacitor and transient time, as presented in Equation (5.1).

5.3.3 Voltage Droop Prediction

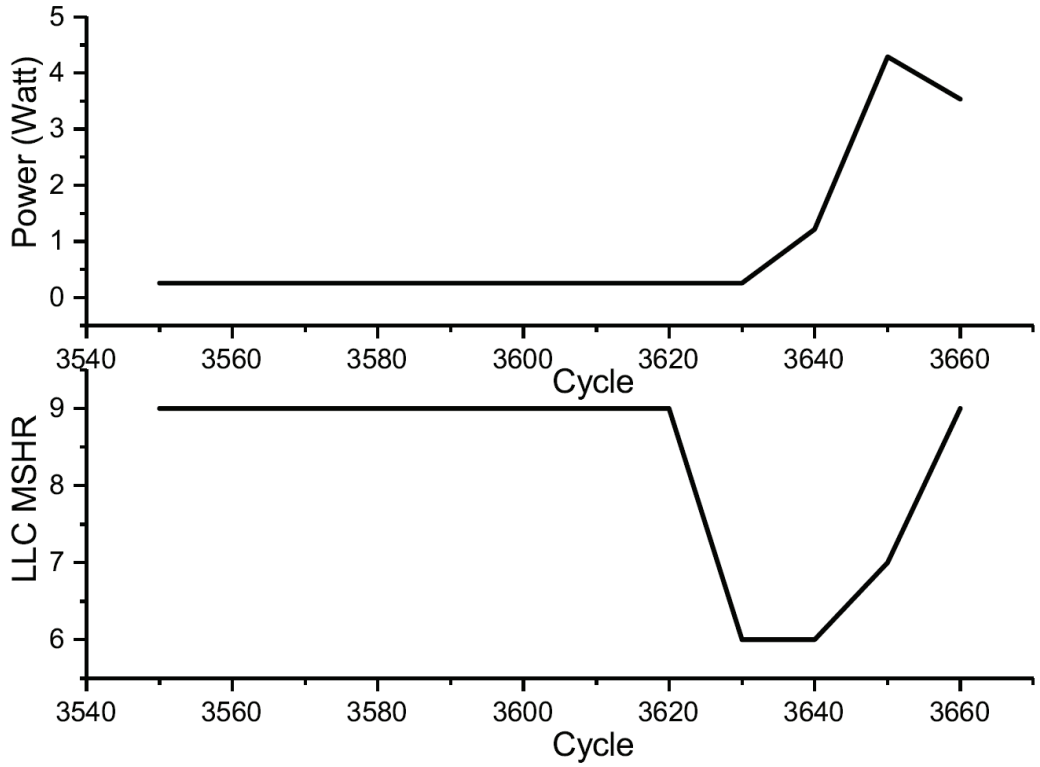
Due to the complexity of power prediction (e.g., non-linear behavior), estimating runtime power consumption in a multicore processor is a challenging task. Many research efforts have delved into learning based methods to improve the accuracy of power prediction [82] [83] [84] over a predefined linear power model. One interesting observation is a strong relationship between microarchitectural events and voltage variations [85] - a relationship we explore and utilize in this paper.

In this work, we propose a learning based prediction system to estimate near future power consumption for droop compensation based on microarchitectural events. The prediction system takes advantage of regression analysis between microarchitectural events and processor power using several alternative machine learning algorithms and proposes an efficient approach for voltage droop prediction.

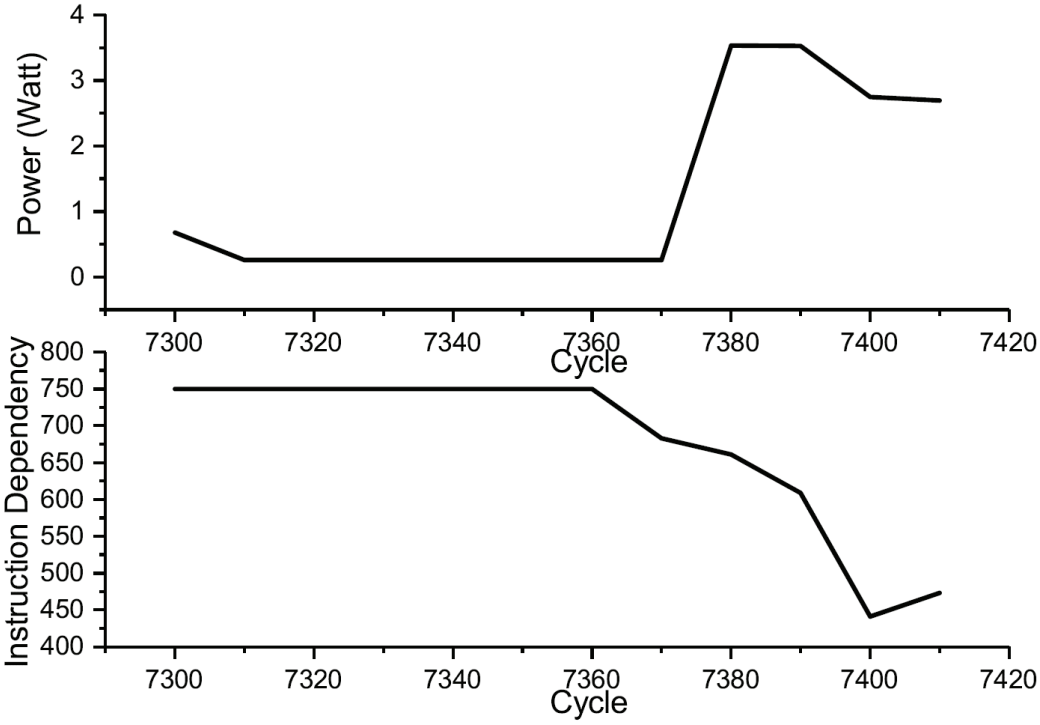
One consequence to note here is the construction the voltage domains. If we compensate via prediction for voltage droop in one part the of the voltage domain controlled by an IVR, the consequence of raising the voltage across the whole domain ought to be recognized. On average, all devices in the domain will now have a higher lifetime operating voltage with consequences for lifetime device reliability. Such complex relationships between power management and device reliability are beyond the scope of this thesis, but reflect useful areas of future research.

To diagnose the power behaviors, we identify critical microarchitectural events responsible for causing load transients. In fact, pipeline stall and recovery leads to a large load transient. Figure 5.14 depicts two types of pipeline activities that cause a significant power increase when a single core executes $raytrace$.

In the first scenario, the last level cache (LLC) miss status handling register (MSHR) occupancy (MSHR capacity 32) reduces from 9 to 7 at cycle 3630, indicating that the data from two LLC misses have been returned from memory. The data reach the pipeline in the next 20 cycles and the execution unit in the pipeline resumes to work. Thus, the



(a)



(b)

Figure 5.14: Power snapshots of the raytrace application in terms of a) LLC MSHR and b) instruction dependencies in ROB

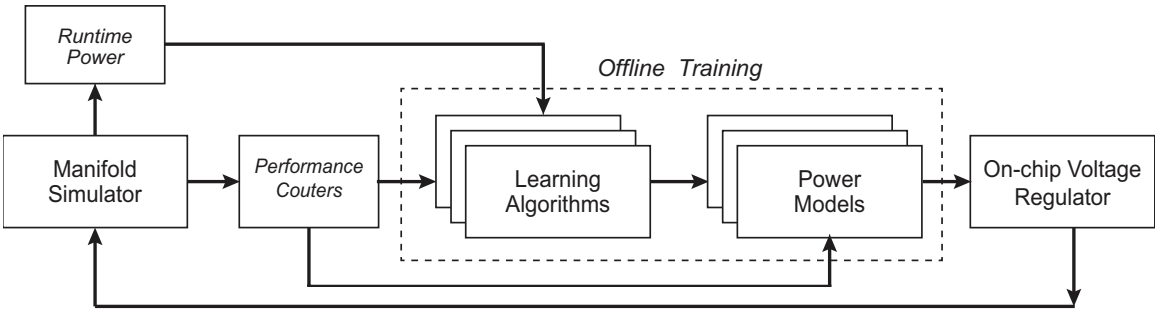


Figure 5.15: The framework of droop prediction

pipeline power increases from $0.25W$ to $4.3W$ at cycle 3650. In the second scenario, we track down instruction dependencies in the reorder buffer (ROB). Starting from cycle 7370, the instruction dependency decreases significantly. Instructions with a high degree of dependencies retire during this period and pipeline can proceed to fetch and allocate instructions to ROB in the next few cycles. Power of the pipeline frontend increases and total power ramps up from $0.26W$ to $3.5W$.

Consequently, we are motivated to develop a prediction system that infers power consumption from pipeline events and cache activities and calculates the new current demand from the predicted power. Note that several types of microarchitectural events are also related and thus may be equivalent in their predictive capabilities.

5.3.4 VDPred Prediction Framework

Figure 5.15 illustrates our framework for learning based development and application of models for voltage-droop prediction, which consists of two phases. In the learning phase, we collect both detailed pipeline information and power consumption at each sampling point, and generate a training data set for off-line analysis. In the inference phase, we place the trained power model with inputs from the core pipeline to predict current variations.

The prediction system utilizes occurrences of microarchitectural events available from performance counters in the processor model to predict future power consumption. The prediction sampling interval is set to $10ns$ according to the droop response time of our

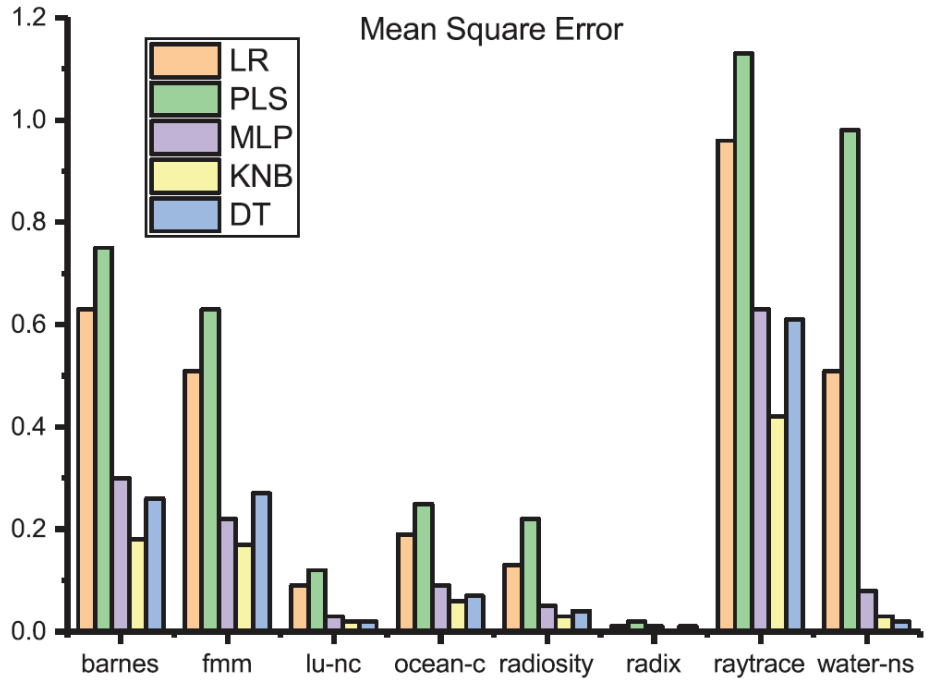
on-chip voltage regulator model. Since VDPred brings a non-linear system to the feedback loop of the voltage regulator with a much higher sampling rate than the loop bandwidth, it will lead to unstable voltage regulation in extreme situations, and the integration of VDPred into a realistic regulator is a major concern for VLSI design engineers. We evaluate VDPred on a simplified IVR model.

The primary goal of this work is to demonstrate the viability of a load transient prediction framework based on higher level microarchitectural events. The simplified regulator model we use to evaluate the potential of the prediction mechanism does not incorporate realistic behavioral attributes of commodity IVRs. Thus, our model establishes the ability to make productive predictions, while practical implementations will require circuit techniques whose exploration while feasible, is beyond the scope of this thesis and is productive area of future architecture-circuit co-design research.

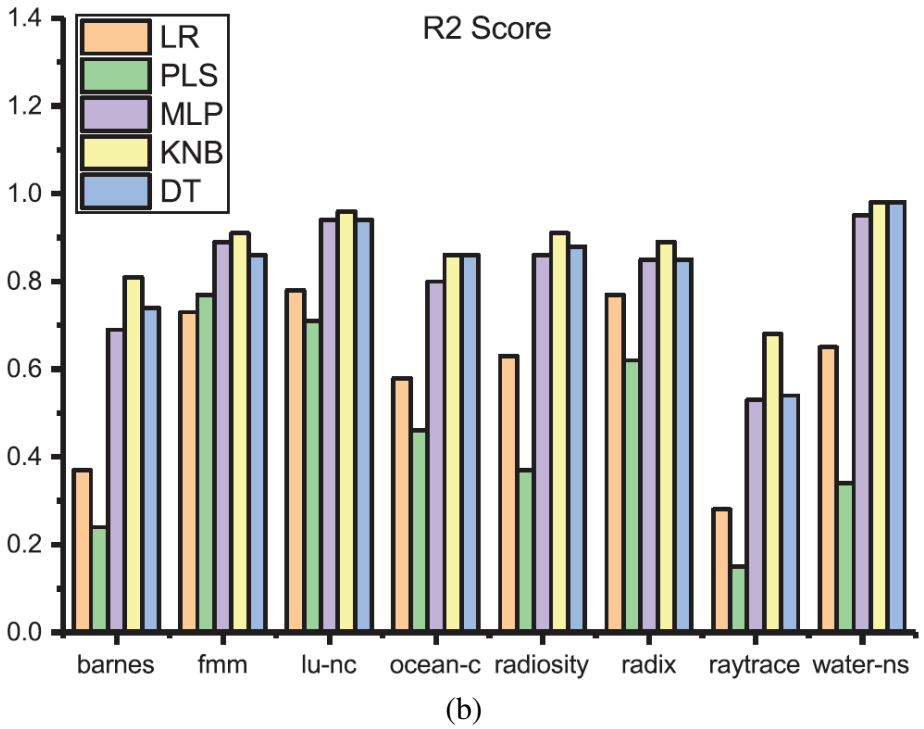
A summary of the performance counters in VDPred is listed in Table 5.4. Group I records pipeline information. We extend the ROB to collect the on-the-fly instruction information that implies current application status including instruction types, age and dependencies. The instruction age records the average cycle of current instructions resided in the ROB after allocation. Group II records the cache activities. We maintain separate performance counters in L1 and LLC to track different types of cache requests and responses.

To capture the relationship between performance counters and power, we conduct offline regression analysis using five general regression models, as listed in Table 5.5, provided by the machine learning library scikit-learn [86]. The offline training set includes two million training points collected by our simulator with a time resolution of $10ns$. We standardize the input features and record the mean and variance for validation.

The power model utilizes the collected performance counters to infer processor power in the next $10ns$ with current performance counters. We validate the trained power models using k-folds cross-validation [87] to estimate prediction errors by plugging the model into the simulator for $500ms$ execution. Results are shown in Figure 5.16.



(a)



(b)

Figure 5.16: Validation comparison of learning models in terms of a) Mean Square Error, and b) R2 Score

Table 5.4: Activity counters in droop prediction

Feature	Description
inst_fetch	number of instructions fetched per sampling interval
inst_retire	number of instructions retired per sampling interval
inst_alu	number of ALU instructions in the ROB
rob_occ	ROB occupancy
rob_age	sum of each instruction age in ROB
rob_dep	sum of each instruction dependency in the ROB
cache_occ	L1/LLC occupancy
cache_miss	missed requests in L1/LLC per sampling interval
cache_mshr	L1/LLC MSHR occupancy
llp2llc	number of requests from L1 to local LLC
llp2peers	number of requests from L1 to L1 peers
all2llp	number of responses to L1
llc2mem	number of requests from LLC to memory
llc2peer	number of requests from LLC to LLC peers
mem2llc	number of responses from memory to LLC

Table 5.5: A summary of our used machine learning models

Learning Model	Description
LR	Linear Regression
PLS	Partial Least Square (Decomposition)
MLP	Multi-layer Perceptron
KNB	K-nearest Kneighbors
DT	Decision Tree

Linear Regression (LR): The model fits a linear model with a minimized residual sum of squares. The R2 score ranges from 0.28 to 0.77 and LR is fairly accurate for memory bounded applications. LR does not work well in compute bound applications.

Partial Least Square (PLS): The model finds a linear relationship between two multivariate datasets. We reduce feature dimension to 5 and capture the internal correlation between features. PLS achieves the worst performance in predicting the power, suggesting a weak correlation between ROB status and cache activities.

Multi-layer Perceptron (MLP): The model constructs a multi-layer network (in our model, it is a 2-layer network) as a non-linear function approximator. We restrict the total number of perceptrons to 50 in the first layer and 15 in the second layer.

Table 5.6: Comparison of the predicted and actual power between learning models

Learning Model	**h0** (P_{old})	**h1** (P'_{old})	**h2** (P'_{new})
LR	0.21	0.27	0.69
PLS	0.32	0.04	0.54
MLP	0.08	-0.04	0.95
KNB	0.03	-0.02	0.99
DT	0.05	-0.02	0.95

K-nearest Neighbors (KNB): The model derives from the k-means algorithm using eight neighboring points. In our experiments, KNB excels other models, especially for compute bound applications (R2 score 0.91 in $radiosity$). It indicates a strong clustering structure in performance counters.

Decision Tree (DT): The model utilizes a non-parametric method to predict power via simple decision rules. It works well (comparable to KNB) in CPU-bounded applications that have a strong non-linear relationship between performance counters and power consumption.

We conduct correlation analysis on the predicted model between the future power P_{new}, present power P_{old} and estimated power P'_{new} and P'_{old}, as shown in Table 5.6.

$$P_{new} = h0 * P_{old} + h1 * P'_{old} + h2 * P'_{new} \tag{5.2}$$

As we can see, the correlation between P'_{new} and P_{new} are smaller in both LR and PLS compared to the non-linear models, indicating a strong non-linear relation between counters and power. Therefore, we utilize a decision tree logic as the basic block in VDPred hardware for power prediction for adequate prediction accuracy and simple hardware implementation.

5.3.5 VDPred System Design

VDPred is a voltage noise smoothing system based on microarchitecture-level learning based droop prediction integrated in the control loop of a voltage regulator. The objective

of VDPred is to effectively minimize the processor power by limiting runtime voltage noise. Figure 5.17 portrays the opportunity of power reduction in VDPred. The dashed regions reflect the power saving in nominal operations when worst-case droop is reduced by $0.12V$.

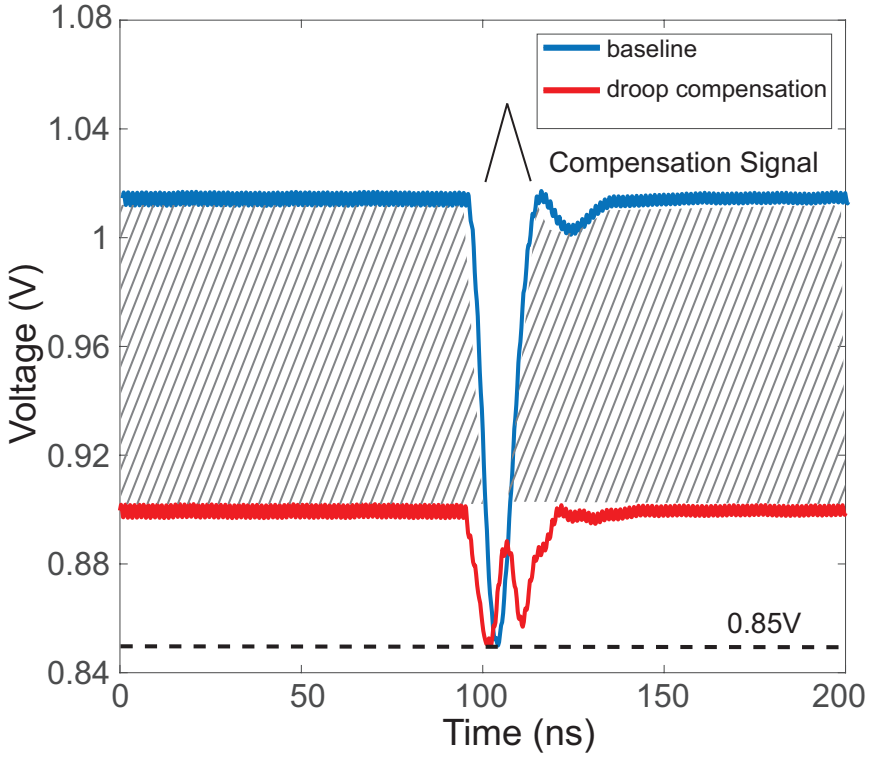


Figure 5.17: Power reduction through guardband reduction

To reduce voltage droops during load transients, reactive control still relies on the detection of droops, which is typically performed by comparing the output voltage to a voltage reference. This process is still slow in terms of core frequency for a agile droop control in processors. VDPred instead accelerates the regulation process through a reliable prediction in processor current demands.

VDPred predicts an increase in the processor current for the next control phase. There can be multiple methods in which this VDPred output is interfaced with the IVR.

1. **Charge injection at the IVR output**: The VDpred output can be used to turn on a transistor parallel to the IVR power stage to supply excess charge to the output node to compensate for the voltage droop. Although this method works perfectly

for a reactive control [81], it is not guaranteed to improve the voltage droop in the present scenario. When the excess charge is dumped at the output node, the output voltage starts to increase; however, as the control loop is closed, the duty cycle is adjusted accordingly to bring the output voltage back to the reference voltage. If the droop occurs during this process, a small improvement in the output voltage can be observed, however, if the droop appears after the regulation point is achieved, the voltage droop won't improve.

2. **Boosting reference voltage**: VDpred output is added to the nominal reference voltage, set by a DVFS controller. The control loop is forced to regulate at the new reference point. If the prediction is accurate, the output drops from a higher voltage when the load current transitions and reduces the absolute droop.

As shown in Figure 5.18, the VDPred framework integrates two levels of droop reduction schemes. The circuit-level system resembles a typical reactive controller constantly sensing the output from the voltage regulator. The microarchitecture-level system implements a droop compensation scheme manipulated by the droop prediction. By leveraging the two levels of runtime information, VDPred can predict and compensate voltage droops and minimize power consumption.

According to Figure 5.13, the drop in supply voltage during a transient is proportional to the increase in current demand. Before a droop occurs, VDPred predicts a current increase in the processor and compensates this possible upcoming voltage droop by a compensation circuit. The compensation circuit temporarily raises the reference voltage in the feedback loop of a voltage regulator when current increases. Equation (5.3) shows the mechanism of droop compensation in VDPred. α is a design parameter of a voltage regulator and $\beta = t/C$ is a parameter depending on the output network of the voltage regulator. Initially, the feedback loop corrects the next voltage V_{new} to V_{ref} based on the difference between reference voltage V_{ref} and V_{old}, given that processor current remains constant. This mechanism will fail if the current changes abruptly between control intervals and thus voltage

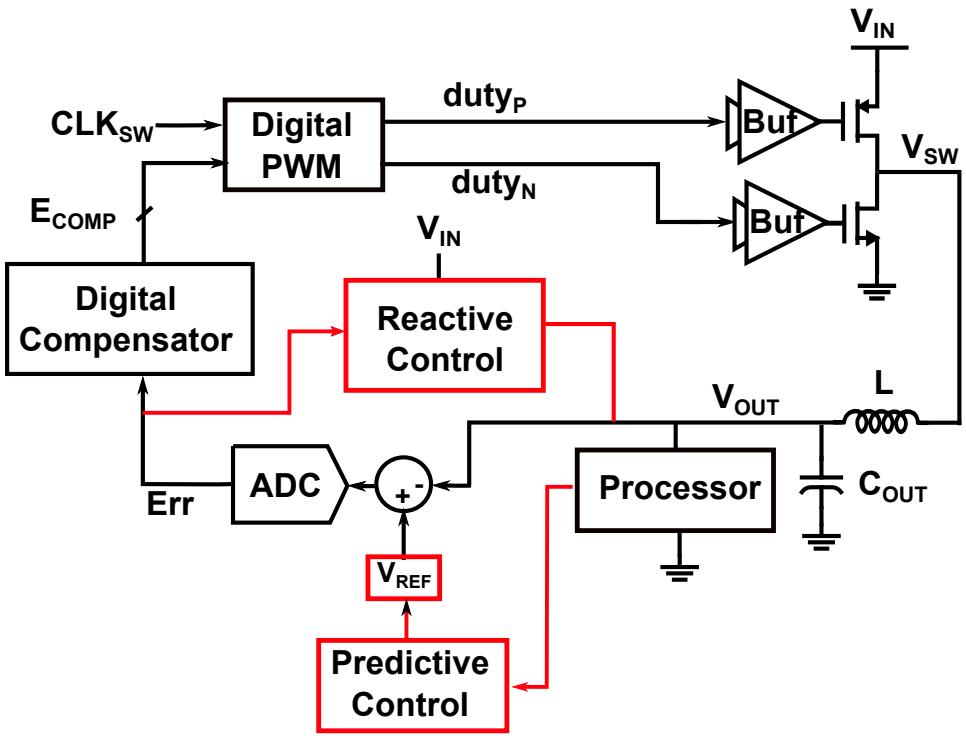


Figure 5.18: The framework model of VDPred

regulation will be out of control. However, to predict current change Δi, we can add a compensation component $\beta\Delta i$ to the feedback signal and update $V_{new}(i + \Delta i)$ to V_{ref}.

$$\begin{aligned} V_{new}(i) &= V_{old}(i) + \alpha(V_{ref} - V_{old}(i)) \rightarrow V_{ref} \\ V_{new}(i + \Delta i) &= V_{new}(i) - \beta\Delta i \rightarrow V_{ref} - \beta\Delta i \\ V_{new}(i + \Delta i) &= V_{old}(i) + \alpha(V_{ref} + \beta\Delta i - V_{old}(i)) \rightarrow V_{ref} \end{aligned} \tag{5.3}$$

We characterize the compensation circuit in VDPred with step current signals as shown in Figure 5.19. Suppose VDPred foresees a current step and enables the pull-up circuit at the current jump. We compare the droops between VDPred and the baseline regulator in Figure 5.12. Results show that the maximum droop reduces by over $0.1V$ ($> 11\%$ reduction in a $0.85V$ reference) across the magnitudes of a current jump from $0.1A$ to $1.8A$.

Consider a scenario of consecutive current events in VDPred. As depicted in Fig-

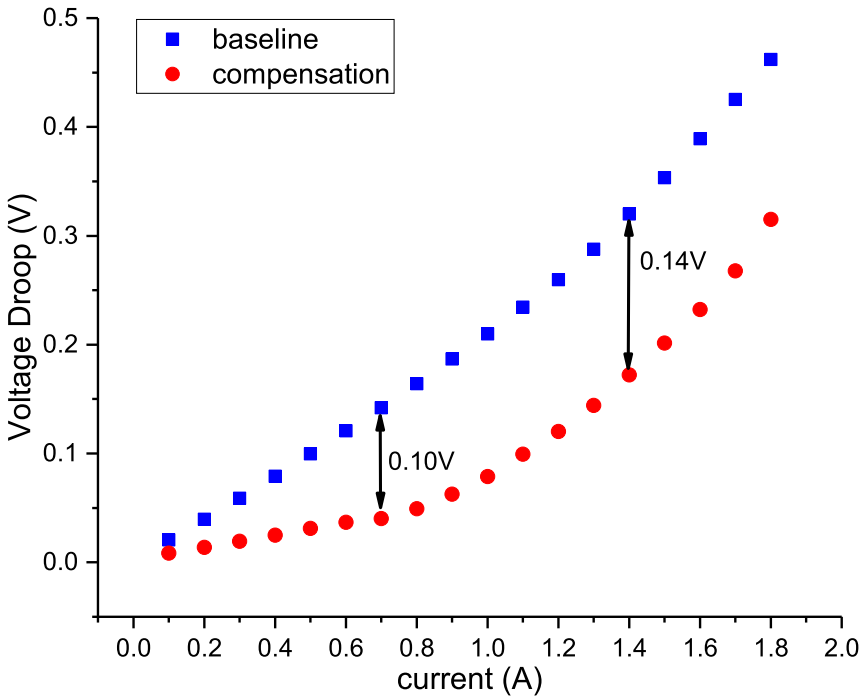


Figure 5.19: Characterization of droop compensation circuit for step current

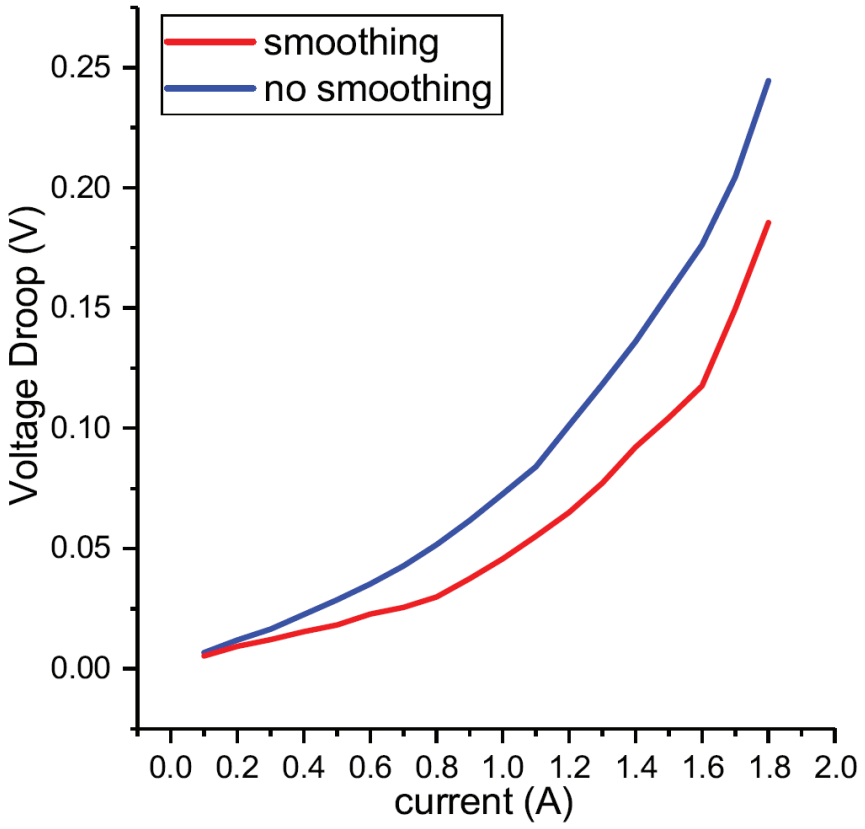


Figure 5.20: Droop reduction for consecutive current events

ure 5.20, two current steps occur at $20ns$ and $40ns$. Since the current between $20ns$ and $30ns$ does not change, VDPred disables the compensation network, which harms the upcoming current jump and worsens the second voltage droop. Instead of lowering the compensation signal, we hold the value of this signal between $20ns$ and $30ns$ to guarantee a smooth transition for consecutive current jumps. The simulation shows maximum $0.03V$ (3.5% reduction) droop reduction for consecutive current jumps.

5.3.6 Misprediction and Handling

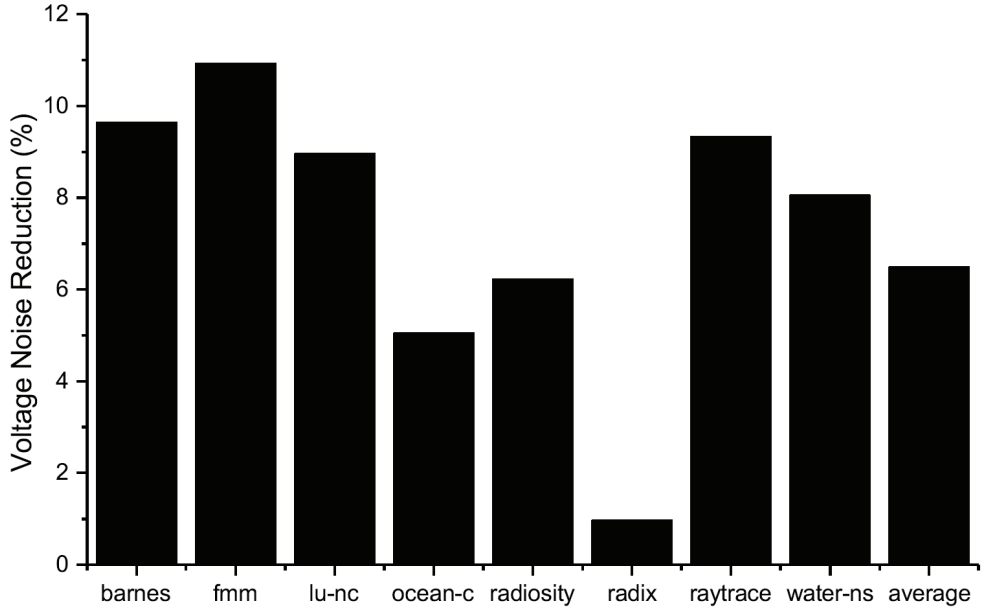
VDPred generates the enabling signal for the compensation circuit from a comparison between present power consumption and predicted future power, which achieves the upper bound of power reduction when the prediction is perfect (i.e., 100% accuracy). Figure 5.21 presents the performance of VDPred in terms of voltage droop, average runtime voltage, and power reduction in our processor model running the SPLASH-2 applications. Results indicate that VDPred can achieve a maximum of 10.9% droop reduction (in fmm) and an average of 14.2% power reduction.

When an error takes place in the droop prediction, it penalizes VDPred's performance in power reduction. Substitute the predicted current change Δi in Equation (5.3) to $\Delta i + i_\epsilon$ (i_ϵ is a prediction error), and the propagated error to the output voltage becomes βi_ϵ, as in Equation (5.4).

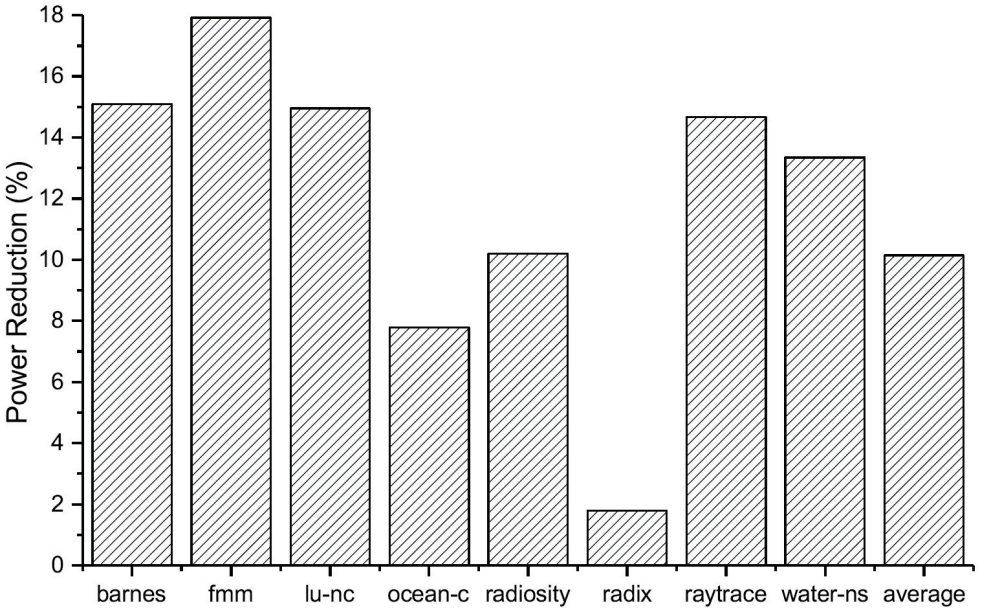
$$V_{new}\prime = V_{old}(i) + \alpha(V_{ref} + \beta\Delta i + i_\epsilon - V_{old}(i)) \rightarrow V_{ref} + \beta i_\epsilon \tag{5.4}$$

To demonstrate the impact of droop misprediction on power, we conduct several simulations that VDPred directly applies the prediction described from the previous five machine learning techniques. Simulation results are shown in Figure 5.22.

When the trained model contains noticeable prediction error, the power consumption would rise. For example, the power consumption with LR predictor in $ocean\text{-}c$ increases



(a)



(b)

Figure 5.21: Reduction of VDPred with perfect prediction in a) voltage noise, and b) average power

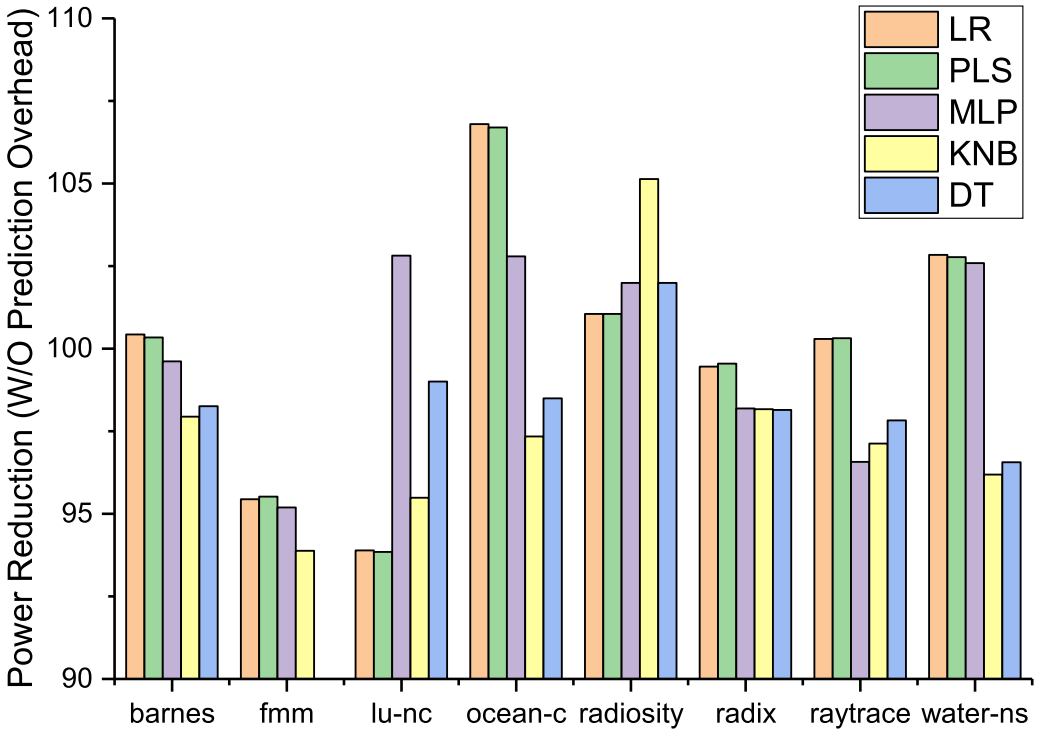


Figure 5.22: Impact of prediction error on power reduction

by over 7%. The total power increases for two reasons. First, a misprediction is a false negative, in which VDPred does not predict a worst-case droop. As a result, the voltage guardband is not reduced and the pull-up circuit contributes to the extra power. Second, a misprediction is false positive, in which VDPred sends a false signal to the pull-up circuit. In this case, the pull-up circuit raises the supply voltage and increases the processor power when unnecessary.

Error in droop prediction could amplify runtime noise margins in extreme cases, in which the prediction error is so large that it alters the expected behaviors of VDPred. Therefore, techniques to handle prediction errors are critical in preventing VDPred to create large voltage fluctuations. Figure 5.23 depicts several schemes in VDPred tackling prediction errors, detailed as follows.

The first scheme is an error resetting structure using a reactive control logic when the voltage difference $V - V_{ref}$ exceeds a given threshold. This situation indicates that the droop prediction contains a large error and VDPred needs to be reset. The control logic

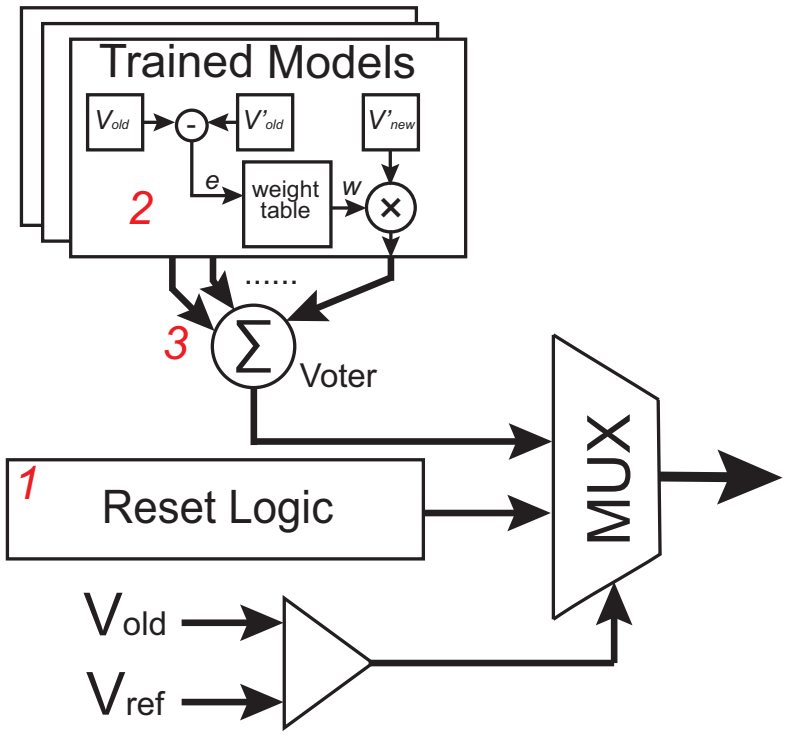


Figure 5.23: Prediction error handling in VDPred

continuously senses the output of the regulator and shutdowns the pull-up circuit and activates a reactive control loop if triggered.

The second scheme is a boosting structure using a prediction error estimating logic based on current system status. Because of the implementation constraints, VDPred implements several weak power prediction models based on decision trees. The basic idea is to ensemble these weak models together and update the weights of these models w_i based on the information of past prediction error $V'_{old} - V_{old}$ so as to minimize the runtime prediction error. In our hardware implementation, we calculate the weights beforehand and store in a table indexed by the prediction error. The final output of VDPred is a weighted linear combination of the decision tree models $\Sigma w_i \times V'_i$.

5.3.7 Hardware Implementation

A accurate predictor implies a long computation latency. In VDPred, the latency in prediction hardware depends on the tree depth and clock frequency. To balance the prediction

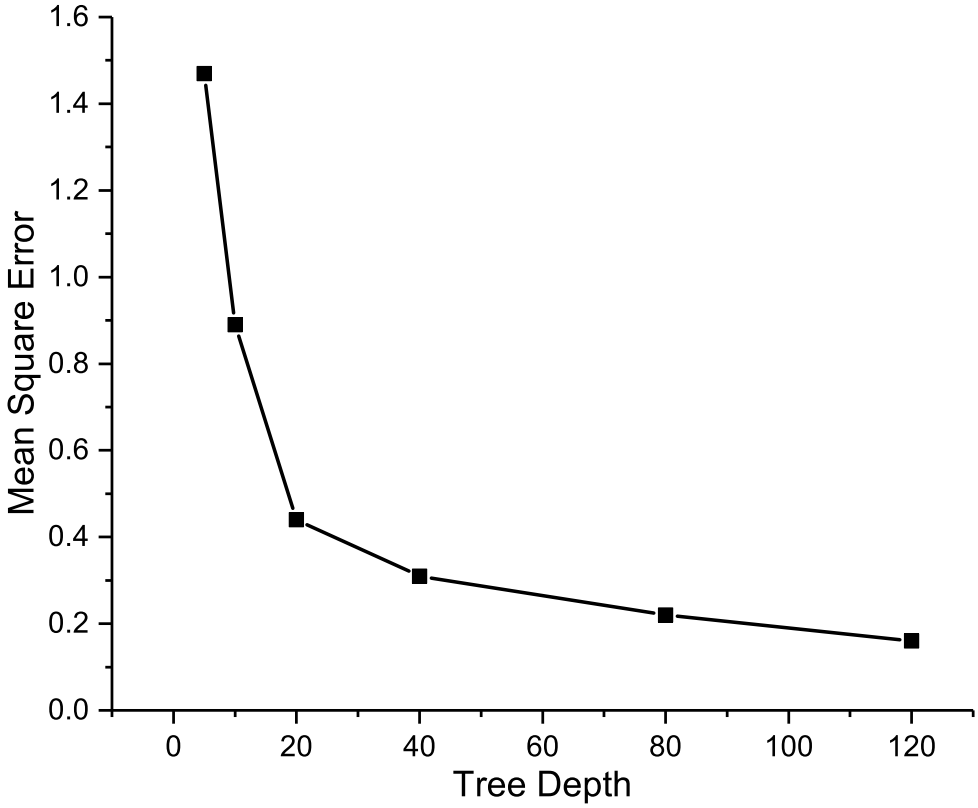


Figure 5.24: Impact of tree depth on prediction accuracy in MSE

accuracy and prediction latency both determined by the maximum depth of decision trees, we characterize the prediction error of a single decision tree model in terms of tree depth running $barnes$, as shown in Figure 5.24. When the tree depth increases above 40, the mean square error (MSE) is controlled.

We utilize the hardware implementation of decision tree in [88] and restrict the maximum tree depth to 40 in our offline training so that the computation result will return within $10ns$ given a circuit frequency of $4GHz$.

The power of decision tree is determined by the total bits processed per cycle multiplied by the energy per bit (assume $2pj/bit$ in recent technology). To suppress the digital noise, data width sets to 8bit (256 levels), and the total power of the decision tree is $0.384W$.

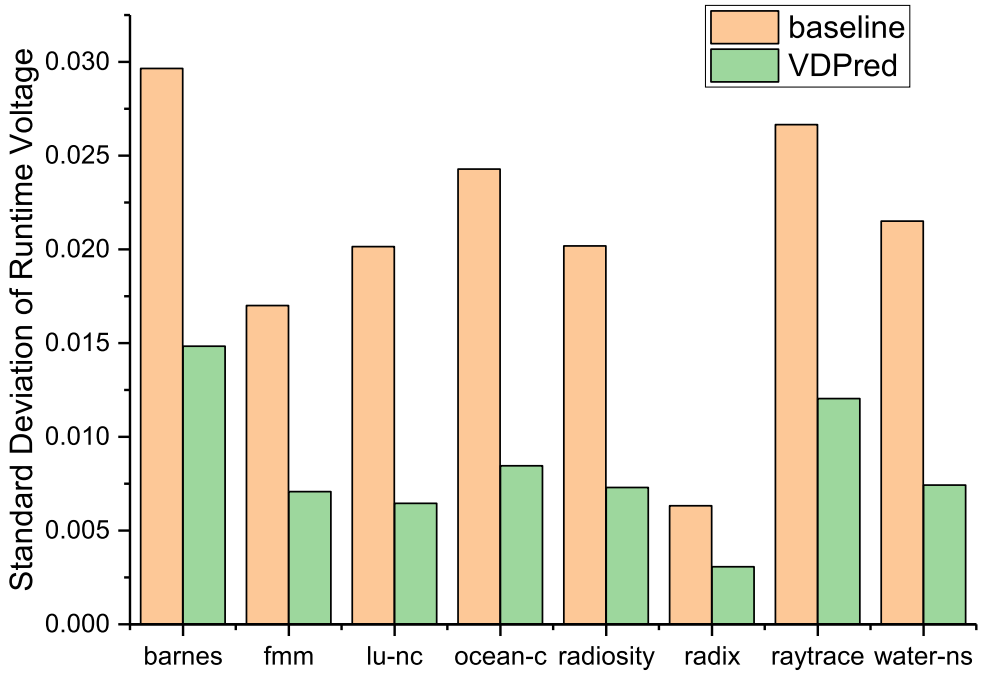


Figure 5.25: Comparison of voltage variance between baseline and VDPred

5.3.8 Resilient Design Exploration

Because VDPred greatly reduces the voltage variance compared to a baseline voltage regulator as shown in Figure 5.25, it can utilize resilient processor architectures to further minimize runtime power.

In a worst-case design paradigm, the voltage guardband sets to $V_{ref} - min(V)$ regardless of the voltage variance. The resilient system such as Razor [89] yet can tolerate aggressive margins and run into recovery when a voltage violation is triggered. In our simulation, we record the number of violation voltage points and add them as performance penalty. We set the voltage guardband in such a resilient system in terms of the standard deviation of runtime voltage for each applications, as shown in Figure 5.26 for $raytrace$.

5.3.9 Results and Analysis

The VDPred simulation consists of 3 major components: a full-system microarchitecture simulator, a machine learning framework and a circuit-level voltage regulator model. We

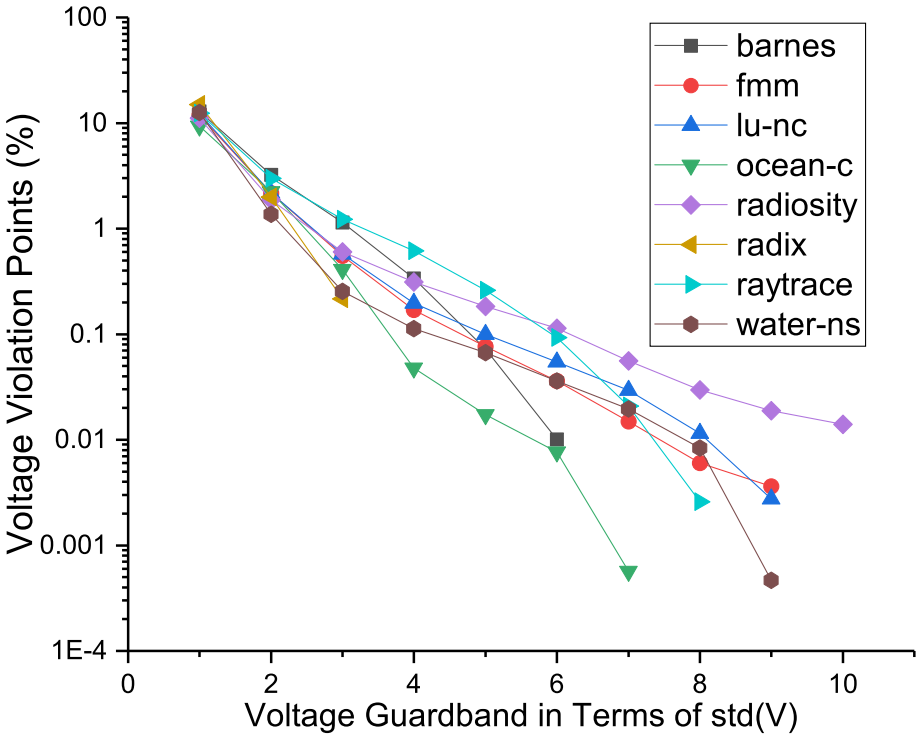


Figure 5.26: Distribution of voltage violation with respect to aggressive guardband

used a time-domain Simulink based model of an inductive IVR based on [90]. The model takes the values of the IVR passives, switching frequency, control loop structure and other parameters and performs a time-domain simulation. For demonstration purpose, an illustrative IVR is used with the following parameters: L=$3.3nH$, C=$25nF$ and FSW=$400MHz$. A type III compensator with two zeros and two poles are used to compensate the power stage of the IVR. Figure 5.27 (a) shows the frequency domain response of the IVR control loop; a phase margin of $43.5°$ and a unity gain bandwidth of $36MHz$ is obtained.

The machine learning framework is built on top of scikit-learn, which implements a set of prevailing algorithms for data preprocessing, classification and regression. The input of the learning framework is the features and processor power extracted from the previous microarchitectural simulator. We then combine the trained model and a voltage regulator model in Matlab to analyze load transients and voltage fluctuations for each application. Finally, we update the voltage and re-run the simulation to update processor power.

One of the main objective of this work is to accurately predict and improve the voltage

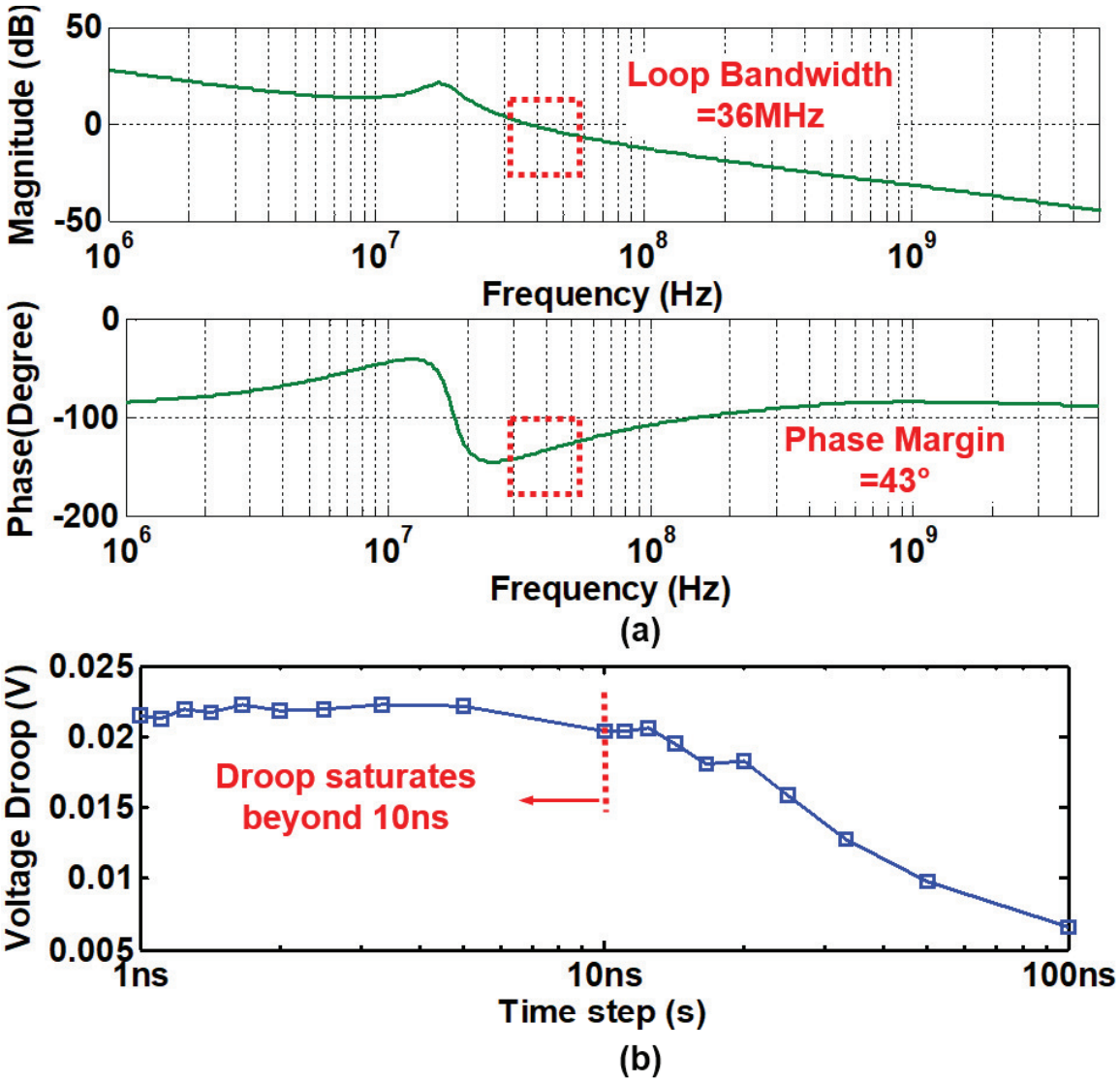


Figure 5.27: IVR Matlab model: a) frequency domain response of illustrative IVR design, and b) droop for different time resolution of PWL current

droop at the IVR output for different processor activities. The simulation framework uses values from a performance counter to generate a piecewise linear (PWL) current waveform consumed by the processor. In an ideal scenario, as the sampling time of the PWL current reduces, the accuracy of the IVR output voltage increases and the Simulink simulation approaches a SPICE simulation. This significantly increases the simulation complexity However, we observe that if the sampling time of the processor current is beyond the IVR loop bandwidth, the maximum droop observed at the IVR output does not change. This is confirmed by observing the voltage droop against sampling time plot, shown in Figure 5.27 (b). Therefore, a time resolution of 10ns is used.

We demonstrate simulation results of VDPred with several voltage regulation systems on a 4-core 3D processor. $Baseline$ implements a baseline voltage regulation without droop compensation. DT implements a voltage prediction system based on decision tree with infinite depth. $React$ implements a reactive control scheme for droop compensation via load current sharing in parallel operations. $React$ enables a bypass circuit to inject current to the processor when the output of the voltage regulator is below a preset threshold. An optimal value of the threshold in our experiments is $0.8V$ for a $0.85V$ reference voltage. $Ideal$ implements a perfect power prediction to compensate voltage droops in the voltage regulator. We present detailed analysis of the results including voltage noise minimization and power reduction.

Figure 5.28 depicts prediction accuracy between DT and VDPred in terms of MSE and R2 Score. VDPred outperforms DT in 6 out of 8 applications, indicating the effectiveness of prediction error handing in the system. For $barnes$ and $raytrace$, VDPred performs rather worse than DT. In these two applications, VDPred fails to capture abrupt power increase as counters changes within the $10ns$ window significantly.

Figure 5.29 (a) compares the results of voltage droop reduction. VDPred achieves an average of 2.6% reduction in voltage guardband compared to $baseline$. With mispredicting handling, VDPred outperforms both $React$ and DT by up to 3%.

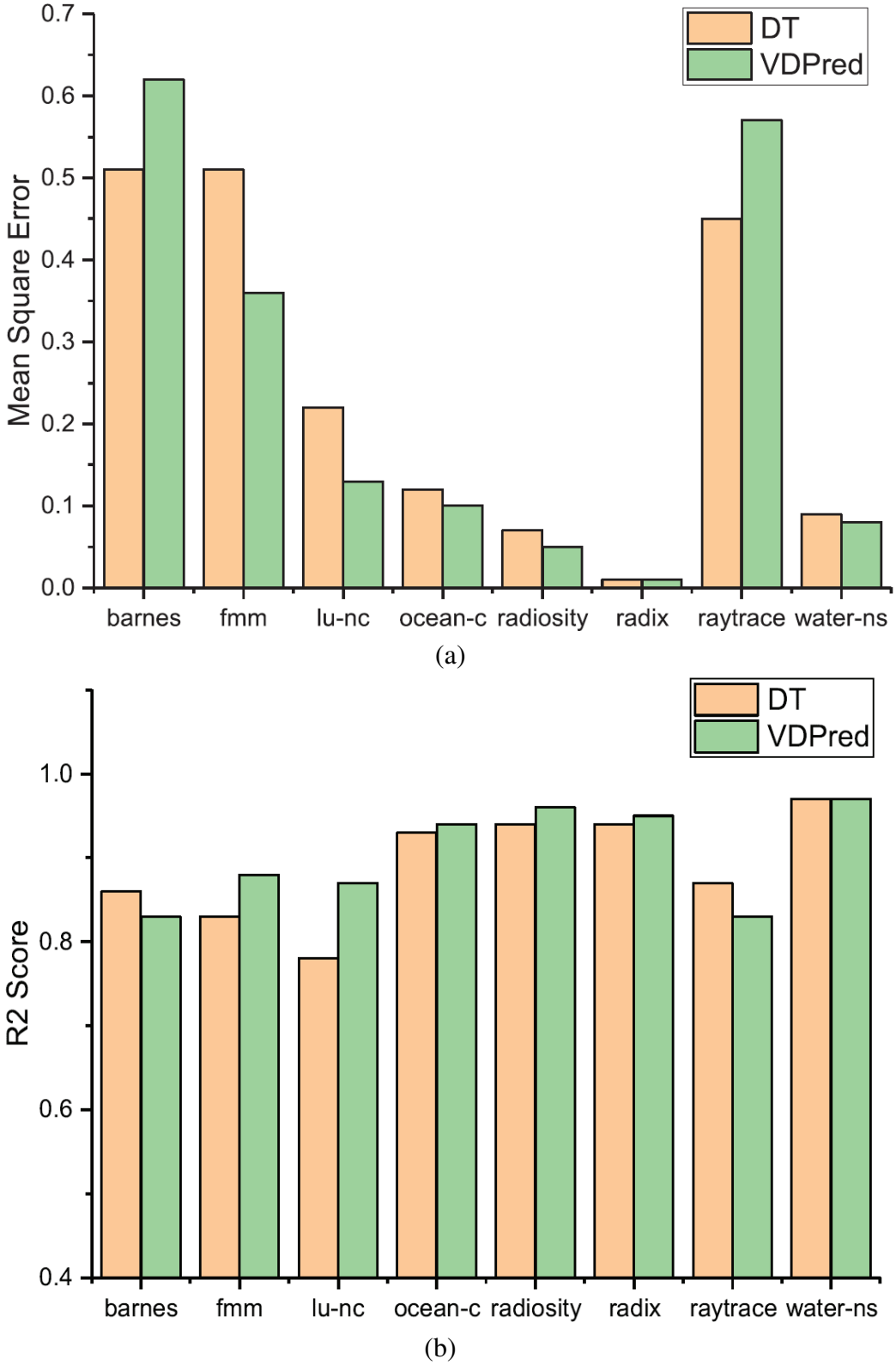


Figure 5.28: Comparison of a) Mean Square Error, and b) R2 score between DT and VDPred

For fmm, VDPred and DT have a relative high prediction accuracy and obtain a roughly 5% improvement in droop reduction compared to $baseline$. VDPred also outperforms $reactive$ in most applications. For $radiosity$ and $raytrace$, $reactive$ outperforms VDPred by 1%-2% as VDPred has a poor prediction performance in these applications. VDPred fails to infer a significant power increase in the pipeline frontend, as the related performance counters change within the $10ns$ time window.

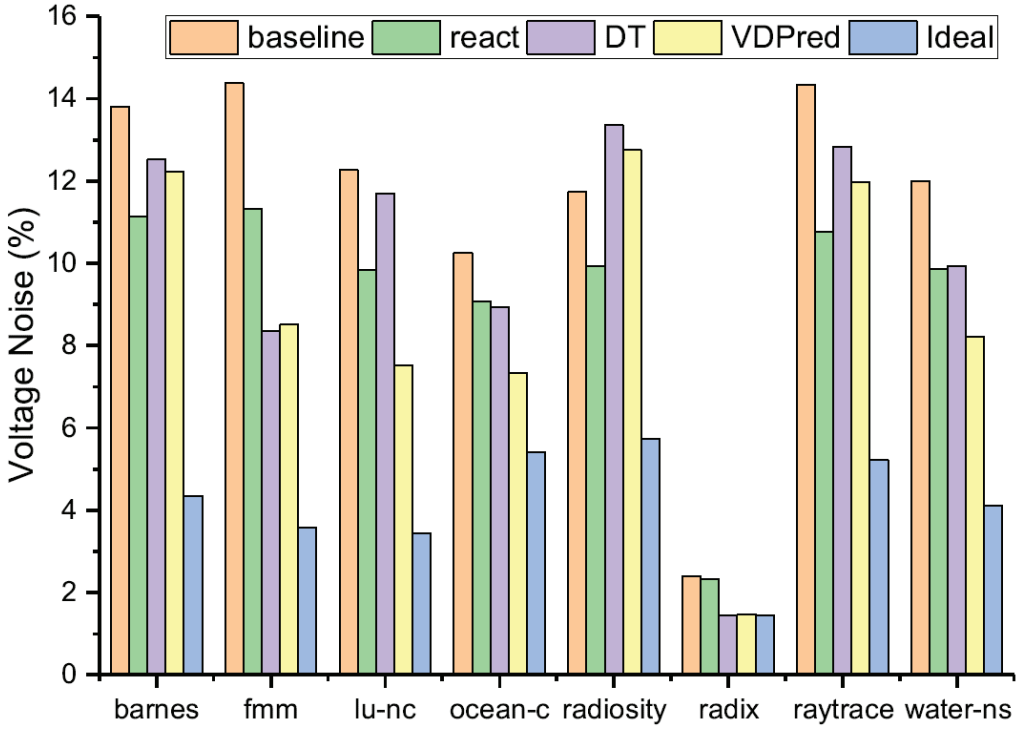
Figure 5.29 (b) presents the results of power reduction. The power reported in VDPred consists of the power consumed in the processor and in the prediction hardware (i.e., roughly 2% overhead). The average power saved in VDPred is 2.5% compared to 1.2% in $react$. Specifically, VDPred obtains a maximum power reduction of 9% in fmm even with prediction power overhead. This is because VDPred manages to predict the worst-case droops when the processor executes the fmm and reduces a 5% noise margin.

When the prediction error is high as in $raytrace$, the power saving in the processor is offset by the power in the prediction unit. In this situation, VDPred introduces nearly 3% overhead. Another interesting observation here is in $radix$. VDPred successfully reduces the voltage margin. However, the processor power executing radix is low and the power overhead from the prediction hardware (constant among applications) penalizes the total power reduction.

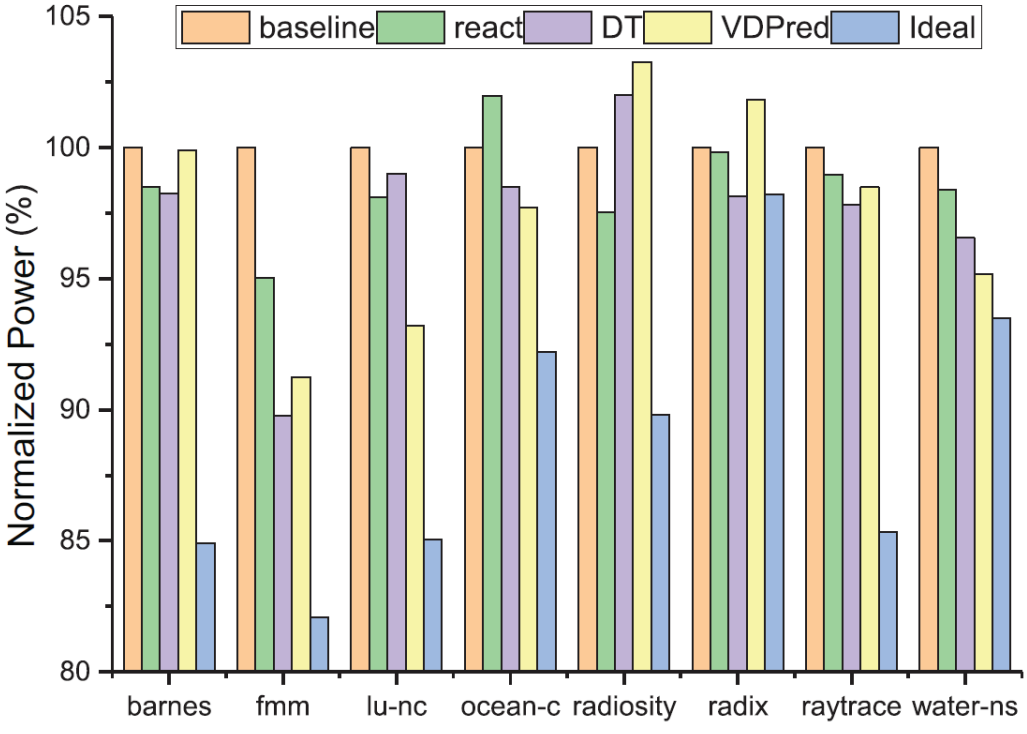
Figure 5.30 presents the results of VDPred applied to a resilient architecture with voltage guardband sets to 6δ. When the recovery cost from voltage violation is 10 cycles, resilient VDPred will obtain extra 5% power reduction over conservative VDPred (over 7% power reduction compared to $baseline$) within 1% performance degradation. If the recovery cost takes 100 cycles, resilient VDPred experiences up to 4% degradation.

5.4 Summary

In this chapter, we focus on minimizing two components of the voltage guardband in the supply voltage of 3D processors through a co-design between circuit models and processor

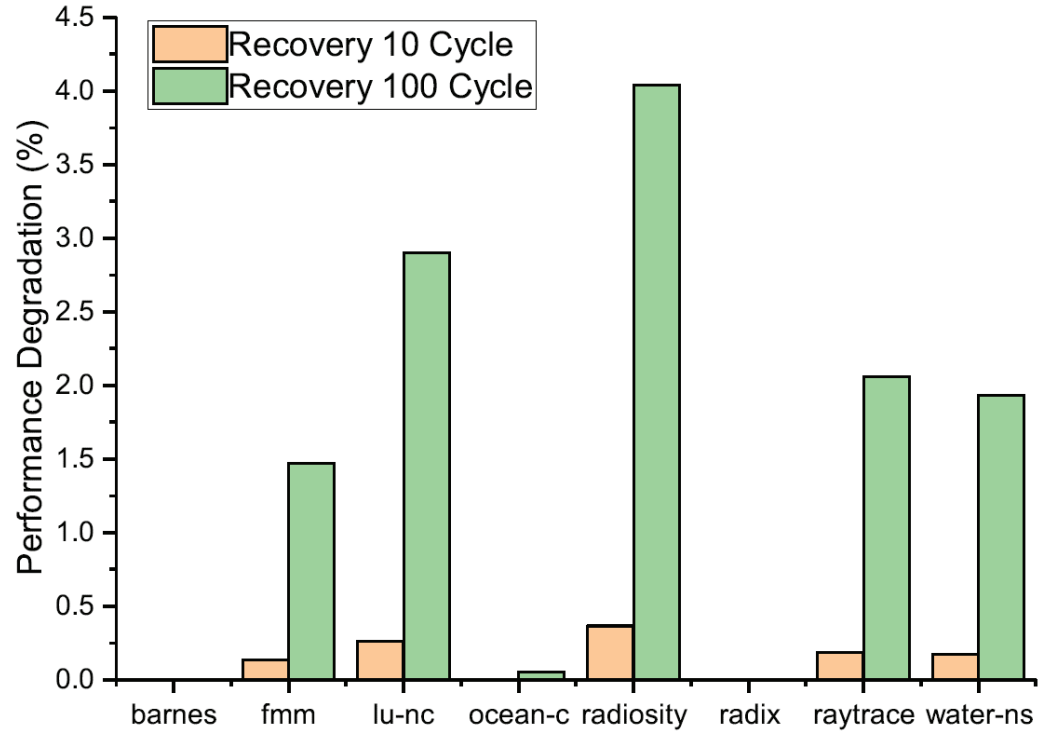


(a)

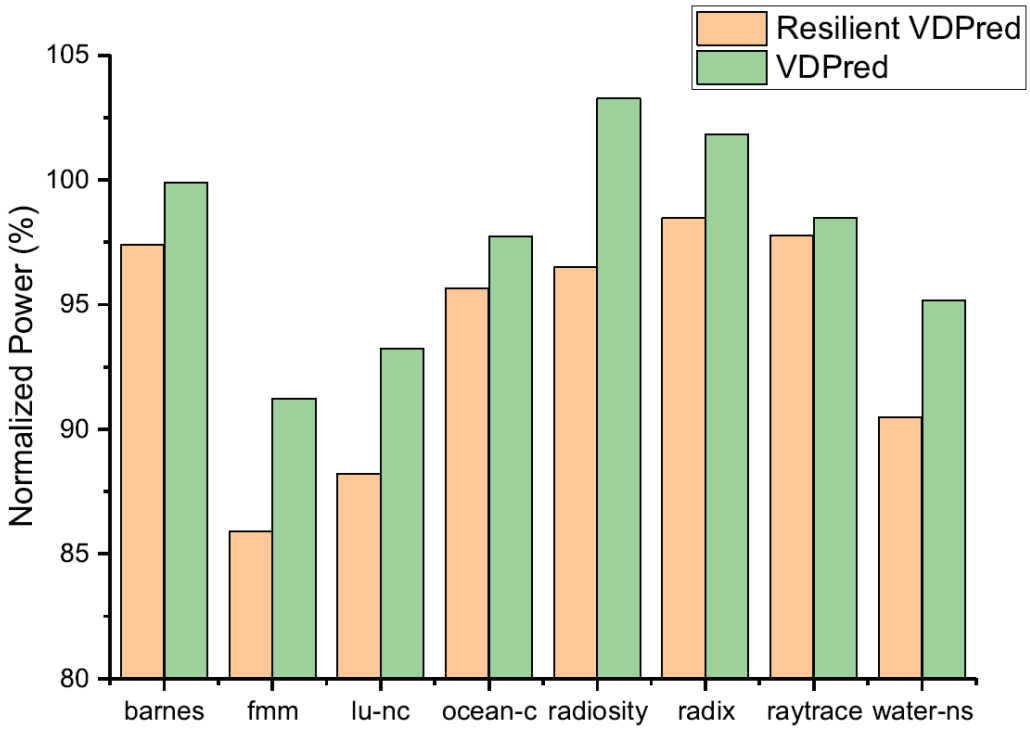


(b)

Figure 5.29: Comparison of a) voltage guardband, and b) power reduction



(a)



(b)

Figure 5.30: Comparison of a) performance degradation and b) power reduction in resilient VDPred design

architecture for improvements in power efficiency.

In the first part of our work, we present a thermal adaptive SRAM LLC, CPM, to improve the energy efficiency and power reduction in the 3D stacked ICs. The key idea of CPM is to construct a bank-level supply voltage regulator that maintains a constant SRAM access time based on the temperature-delay dependence of SRAM LLC. The novelty of CPM lies in that the voltage scaling of LLC is controlled by the temperature and the power of its associated cores, directly addressing the new thermal challenges of 3D ICs. We evaluate the system performance, power/energy consumption, and the energy efficiency of the proposed adaptation technique to both 2.5D and 3D packaging. The simulation results show up to 30% reduction of the peak power and 27% saving in energy consumption of the SRAM cache, compared to the conventional worst-case SRAM design. The memory bounded applications are most benefited from the CPM mechanism. The EPI of the 16-core processor is improved on average by 5% in the 2.5D packaging and 8% in the 3D packaging. The co-design approach to the adaptation SRAM structure indicates potential opportunities to build an high performance, power and energy efficient system using 3D stacked IC technology.

In the second part of the work, we propose a droop compensation system VDPred to minimize power consumption by power prediction based on microarchitectural events. Unlike prior research that focused on circuit events (i.e., load current) to improve transient responses, our approach integrates a comprehensive control mechanism in on-chip voltage regulators that includes both circuit-level reactive control and microarchitecture-level droop prediction. The key insight is that microarchitecture events, that is, cache events or re-order buffer events, *collectively* form accurate predictors of load transients. We explore several learning models for the off-line construction of voltage droop predictors from microarchitectural events. Compared to conventional techniques, we demonstrate that our approach achieves improved guardband reduction and consequently improvements in power and energy efficiency. In particular we are interested in understanding why some learning

models create better predictors and thereby gain a deeper understanding of the power consumption behaviors of multicore processors. This combination of microarchitecture and circuit models forms a circuit-architecture co-design paradigm for processor voltage regulator modules to achieve power efficient processors by (1) enhancing droop predictions with learning based algorithms, (2) designing control logic based on regulator characterization and (3) analysis of the design space of droop prediction for on-chip voltage regulators.

CHAPTER 6

CO-DESIGN OF PROCESSOR ARCHITECTURE AND 3D PACKAGING

6.1 Introduction

The technology of 3D IC enables DRAM stacks inside a single package leading to massive data bandwidth and refactored memory latency path. As such, the paradigm of 3D processor design need to emphasize the importance of scalability and (energy) efficiency with respect to the new features and challenges brought by 3D packages. This chapter focuses on a co-design paradigm of processors between architecture and 3D packages from two perspectives. In the first part of this chapter, we address the problem of pin stress by a 2-tier 16-core processor structure with eDRAM LLC to offset the off-chip data requests, which relaxes the number of signal pins needed by a package. In the second part of the chapter, we characterize the behaviors of 3D in-order cores, and promote a 3D heterogeneous multi-core processor that maximize the power (and energy) efficiency for given constraints (i.e., area, power, and cooling capability). A heterogeneous processor consists of a complex out-of-order core and multiple in-order cores. We propose a thread utility based scheduling in this heterogeneous design to switch workload between in-order and out-of-order cores to further improve system performance. These two researches demonstrate the necessity of the co-design between processor and 3D packages to enhance energy efficiency and improve system performance.

6.2 Pin Stress and Short-Stack Architecture

6.2.1 Motivation

As computing systems move to extreme scale, the number of cores integrated into the package will dramatically increase, exerting pressure on the pin bandwidth between the

on-chip cores and off-chip memory system. According to a recent study by Phillip et al. [42], the total pin counts doubles every six years on average across different ranges of processor design. In order to maintain constant resistive loss per supply pins, the required supply pins grows as the square root of supply current Equation (6.1). As a result, supply pins will take up a large proportion of the total number of pins in a package and expose increasing pressure on signal pins available for the package.

$$\frac{dPower_{pin}^2}{dI_{supply}} = \frac{d((\frac{I_{supply}}{Num_{pin}})^2 R_{pin})}{dI_{supply}} = \frac{2I_{current}R_{pin}}{Num_{pin}^2} = const. \tag{6.1}$$

This slow growth of the number of pins per package coupled with increasing device densities is leading to decreasing off-chip memory bandwidth per core which in turn leads to reductions in system level performance, especially for memory intensive applications. Possible solutions to address this problem include either reducing current demands of the processor package (i.e., integrating on-chip voltage regulators) or reducing the off-chip memory bandwidth requirement. In this work, we present a 2-tier 3D processor structure called the Short-Stack to minimize the demand for pin bandwidth with a FinFET-based embedded dynamic random access memory (eDRAM) configured as the last level cache (LLC). It is also demonstrated to be competitive with 3D architectures using stacked DRAM (i.e., Micron's Hybrid Memory Cube) in terms of manufacturing complexity and TSV reliability.

The Short-Stack processor integrates both the core and LLC ties vertically together into a single package using face-to-face bonding. With the abundant LLC capacity offered by eDRAM, large inter-tier bandwidth and low-latency communication can compensate the performance loss caused by the limited off-chip data bandwidth. The benefits of deploying the Short-Stack structure come from two sources. First, replacing the SRAM with eDRAM cells in LLC increases the cache capacity by roughly $2\times$ with little performance loss [91]. For a typical LLC design optimized for bandwidth, increasing the cache capacity and associativity will reduce off-chip data requests. Second, the latency decreases between the

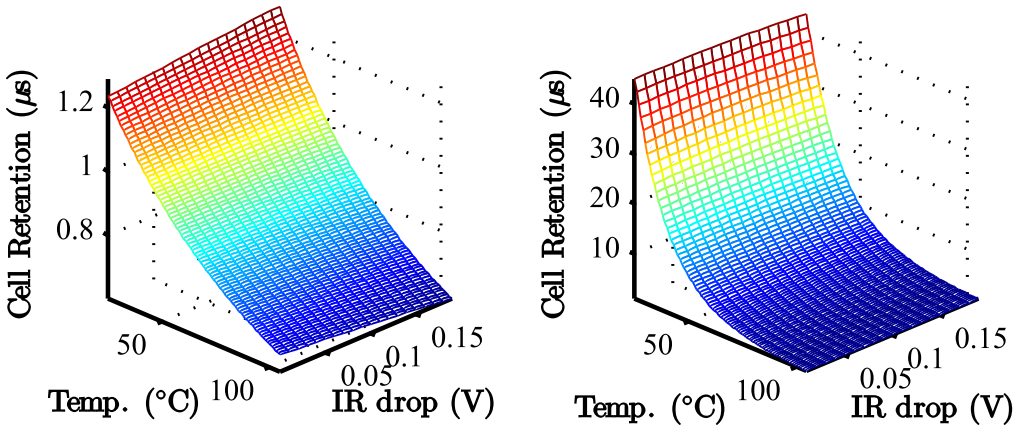
core and LLC tiers in a 3D package, which improves the overall system performance and shorten the performance gap between planar and stacked DRAM systems.

6.2.2 eDRAM Cell Modeling

In advanced process technologies, SRAM becomes increasingly difficult to design due to read-write access contention and eDRAM is emerging as a rising alternative to the mainstream six-transistor (6T) SRAM design [92]. As the 2.5D interposer integration moves on-board memory into package and improves memory bandwidth, eDRAM has made its way into the commercial micro-processor as the in-package last level cache. Compared to traditional SRAM design, the eDRAM cells are more compact because of fewer transistors in the cell design reducing the required cell area by nearly 50%. The associated leakage power is also reduced and the absent of access contention between read and write accesses also improves the voltage margin in eDRAM cells.

However, system-level improvements from eDRAM integration come at the cost of more rigor physical tolerance, especially the reliability precaution such as thermal data retention. Contents of eDRAM cells expire by design due to the lack of an equivalent static retention mechanism and cell refresh is a constant upkeep to maintain required memory states. Further, the availability of eDRAM cell directly correlates with the refresh rate and cell temperature consequently. The dynamic behavior in the eDRAM cells makes the physical condition modeling more critical for system-in-package (SiP) integration. To understand the performance and robustness of eDRAM cells in 3D package integration, we deploy an in-package simulation framework by Wen Yueh [5] based on a distributed RC circuit model in HSPICE, detailed in Appendix B.

We evaluate two technology nodes of memory devices: planar $45nm$ and $16nm$ FinFET models from [93] [67]. As demonstrated by the simulation results, eDRAM cells are comparable to 6T SRAM cells in access delay (i.e., read and write). Moreover, Figure 6.1 highlights the relationship between leakage and thermal outcomes. Compared to the planar



(a) (b)

Figure 6.1: eDRAM cell retention time in (a) planar, (b) FinFET

design, the leakage slop ramps up faster in FinFET Designs, suggesting that the temperature is a critical factor to eDRAM operations. As the retention time of eDRAM has great impact on cell availability, runtime power, and system bandwidth, simulation results imply the use of advanced cooling and thermal-adaptive refresh in eDRAM systems is essential to sustain overall system performance.

6.2.3 The Short-Stack Processor

The Short-Stack processor, depicted in Figure 6.2, consists of a core die and an LLC die. Both dies are stacked in a 3D 2-tier structure using $16nm$ technology. In the simulation model used in this paper, the bottom tier implements 16, x86 out-of-order cores based on Intel Nehalem processor, each with a private $16KB$ L1 instruction cache and $32KB$ data cache. Each core has five components: FE (pipeline frontend and L1 instruction cache), SCH (Out-of-Order scheduler), INT (integer unit), FPU (float-point unit) and DL1 (L1 data cache). The top tier is the proposed eDRAM LLC partitioned into 16 banks. Each cache bank contains $2MB$ capacity, and the content is shared among processors. The memory controllers are also integrated at the corners of the top tier. When there is an off-chip

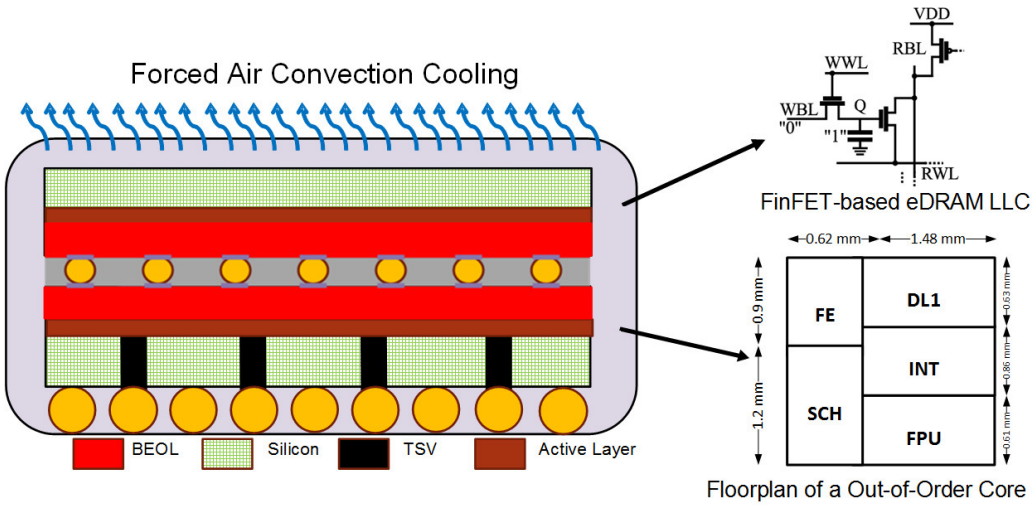


Figure 6.2: The Short-Stack structure using FinFET-based eDRAM LLC

memory request, it will be passed down through the bottom tier via TSVs by the on-chip memory controllers.

Each tier of Short-Stack contains three layers: BEOL layer, active layer, and silicon base. The BEOL layer obtained by lift-off process is used for bonding and routing with a thickness of $25\mu m$. The device layout lies in the active layer that generates the heat. The thickness of the active layer is $10\mu m$. The silicon base layer represents the silicon substrate with a thickness of $25\mu m$. TSVs are embedded in the silicon base layer of the processor die to establish the off-chip memory communication. The Short-Stack is placed on a bismaleimide triazine (BT) substrate through a silicon interposer. The BT substrate is attached to the printed circuit board using solder ball array. We assume forced air convection on top of the chip stack with a heat transfer coefficient of $100W/m^{2\circ}C$.

We evaluate four system configurations with a fixed silicon area. The baseline 2D configuration places the cores and LLC in the same planar floorplan. The ss-SRAM and ss-eDRAM are both Short-Stack configurations with LLC implemented in SRAM and eDRAM respectively. The 3D system model stacks the main memory on top of the processor, which serves as the upper-bound of system performance. LLC of the 3D processor uses eDRAM instead of SRAM due to leakage concerns at high temperature [3].

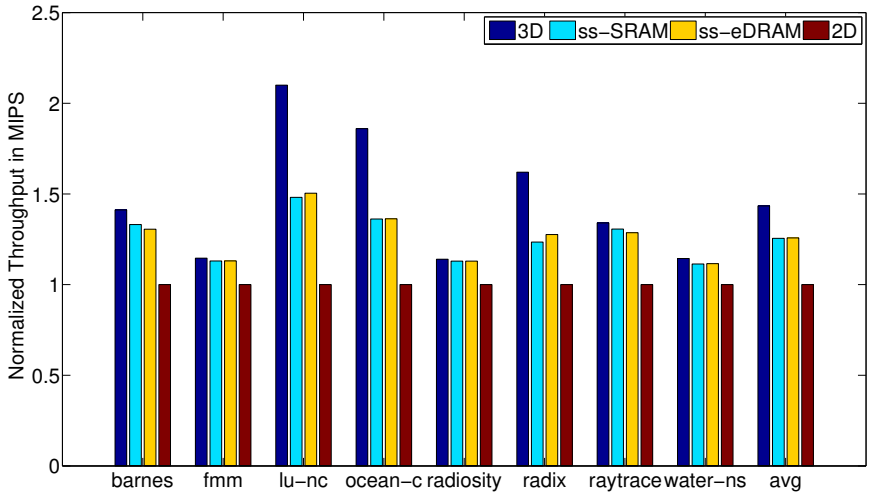


Figure 6.3: Performance comparison between the baseline and Short-Stack systems [5]

6.2.4 Results and Analysis

We evaluate the Short-Stack processor with SPLASH-2 applications, which are fastforwarded to the region of interest and executed till completion. The timing model interacts with Energy Introspector for power and thermal analysis in every $10\mu m$.

Figure 6.3 presents the throughput comparison. The 3D processor has the highest performance gain because of the high memory bandwidth and low access delay, improving throughput by 43.4% on average compared to 2D baseline. ss-SRAM and ss-eDRAM also show an average MIPS improvement of 25.5% and 25.8% respectively. The increased LLC capacity of the ss-eDRAM compensates for the access loss from eDRAM cell retention (i.e., cell refresh). For example, the throughput gain of ss-eDRAM is 27.6% compared to 23.6% in ss-SRAM when executing radix because of the improved LLC hit rate in ss-eDRAM.

The power consumption is proportional to system throughput, as shown in Figure 6.4. The power increase is significant among memory bound applications such as $lu\text{-}nc$, $ocean\text{-}c$ and $radix$. These applications are accelerated by at least 25% using the Short-Stack structure. Meanwhile, the power consumption of the computation bounded application increases

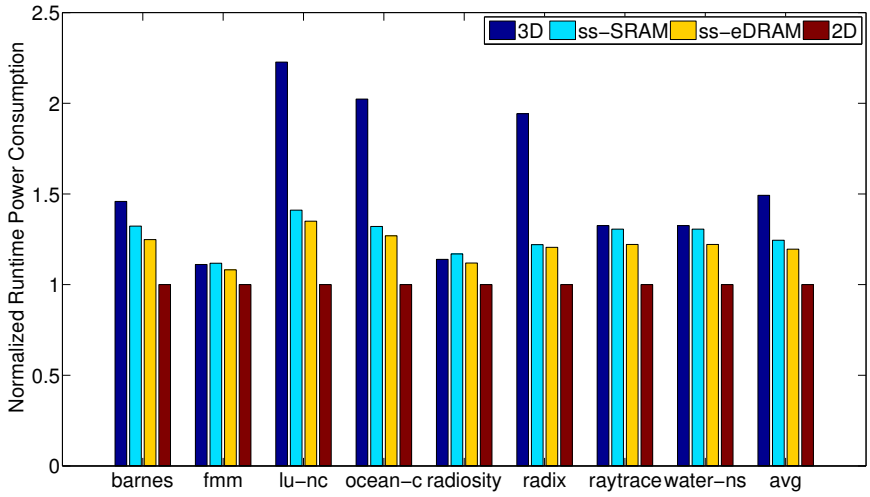


Figure 6.4: Power comparison between the baseline and Short-Stack systems [5]

by around 10% in Short-Stack. The average power increase of ss-SRAM and ss-eDRAM are 24.4% and 19.5% respectively. The ss-eDRAM has overall 5% less power consumption than ss-SRAM as the leakage power of the eDRAM LLC is approximately 55% less compared to the SRAM implementation.

Although the average power increases in both 3D and Short-Stack, the total energy consumed by the entire system is reduced, as depicted in Figure 6.5. The execution time of each application is largely reduced and thus static energy is saved. The 3D processor achieves the highest energy saving of 7.4%, as the execution time reduces by 30%. ss-SRAM and ss-eDRAM get an energy saving of 2.9% and 5.7% respectively. We notice that the energy is increased by 1.3% in ss-SRAM when executing $water\text{-}ns$. $Water\text{-}ns$ runs at higher temperature and the increase in leakage power offsets faster execution.

We measure energy efficiency in energy per instruction (EPI) as shown in Figure 6.6. System EPI reduces both in 3D and Short-Stack. The 3D structure reduces the average EPI by 7.4%, while the Short-Stack processor reduces EPI by 2% and 3.7% respectively in ss-SRAM and ss-eDRAM. The memory bound applications achieve over 5% EPI reduction in Short-Stack, as the performance gain surpasses the power increase. Computation bound

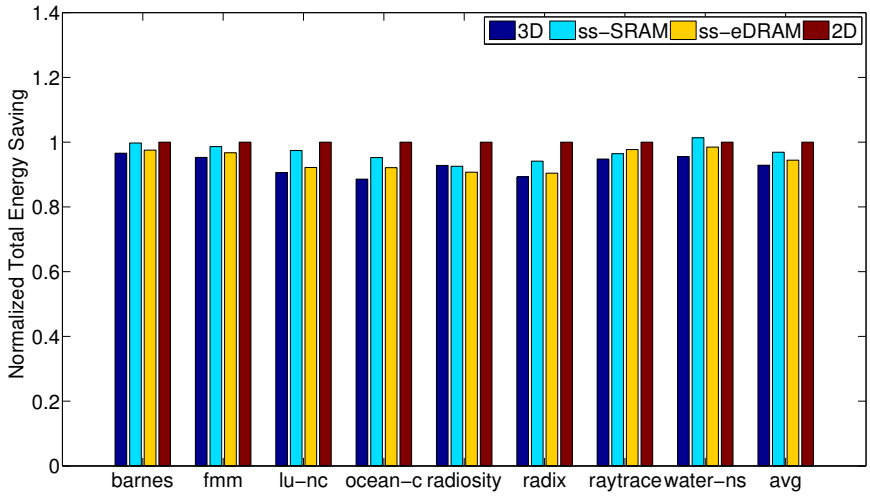


Figure 6.5: Energy comparison between the baseline and Short-Stack systems [5]

applications benefit less for the limited memory interactions. When temperature is high, leakage power of SRAM cells increases significantly. As a result, ss-SRAM suffers from efficiency degradation when executing radiosity and water-ns and system EPI increases by 2.2% and 0.6% compared to the 2D baseline.

6.3 Exploring Power Efficiency in 3D Heterogeneous Multi-core Design

6.3.1 Motivation

To mitigate the "power wall" problem in multi-core processors, heterogeneous design have been studied [94] [95] [96] [97], which exploits runtime performance and power variations. In this section, we motivate a heterogeneous 3D processor design based on a comprehensive analysis between in-order and out-of-order cores in a 3D package, detailed in the following subsections. Floorplans of both cores are given in Figure 6.7. We argue that such asymmetric design is especially suitable for 3D processors in that the power consumption is a major concern in 3D ICs.

The out-of-order cores are based on the Intel Nehalem processor. It has a dimension of $2.16mm \times 2.16mm$ and TDP is $6.25W$. The in-order cores resemble a typical 5-stage

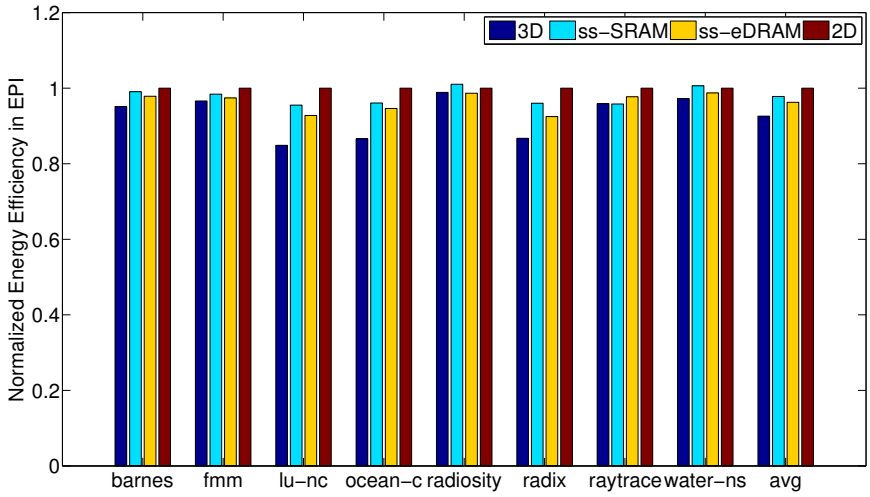


Figure 6.6: Energy efficiency comparison between the baseline and Short-Stack systems [5]

in-order pipeline with a dimension of $1.07mm \times 1.07mm$ and TDP is $1.5W$.

Besides, the stacked DRAM enabled by 3D packages improves the memory latency path, which significantly improves the system performance of in-order core configurations. Figure 6.8 compares the system performance of a single in-order core between a 2D and 3D configuration in terms of MIPS. Compared to the 2D planar design, the in-order core in a 3D system improves 16% system performance in average.

Next, we compare the runtime variation of system performance between a single in-order and out-of-order core, as demonstrated in Figure 6.9. In-order cores have less MIPS variations running computational bounded applications, while out-of-order cores get less variations running memory bounded applications. The reason for this is that out-of-order utilizes ROB to hide the memory latency yet re-ordering instructions could lead to performance fluctuation when memory access is not the performance bottleneck (as in computational bounded applications).

Base on these observations, it is viable to integrate simple in-order cores into 3D architecture as a replacement of out-of-order cores for power benefit. In this section, we explore the design space of 3D processors that consist of a few in-order and an out-of-order cores

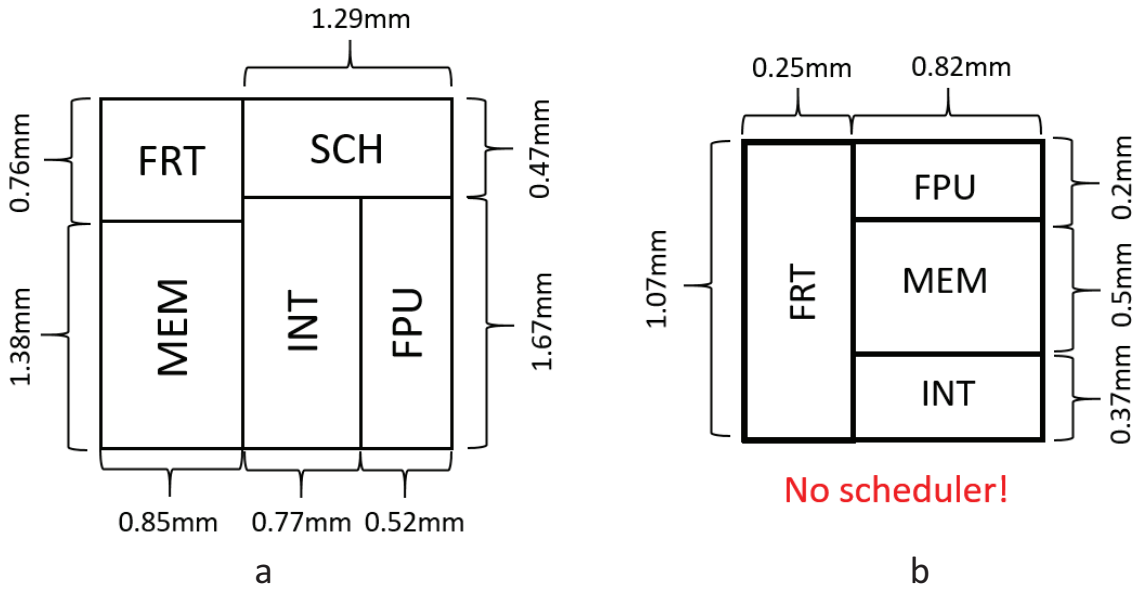


Figure 6.7: Processor floorplan of an (a) out-of-order core, and (b) in-order core in a 16nm technology node

with a shared L2 cache, and examine both conventional multi-core and graph applications running in such asymmetric systems and motivate the use of fine-grained thread scheduler 3DSched to improve the overall power efficiency and system performance.

Specifically, we look to improving the power efficiency of a 3D tiled multi-core processor as depicted in Figure 6.10. The heterogeneous design consists of a single out-of-order core (OOO) and three in-order cores. We address the problem based on a fine grained migration of application threads that identifies the runtime phases of each thread and makes scheduling decisions based on the runtime information. The shared cache design in the asymmetric processor reduces that migration cost.

6.3.2 Execution Behavior Analysis

The motivation of an asymmetric processor stems from the inherent variation in performance-power profile of workloads. In our research, the workloads consists of 20 applications collected from SPLASH-2, PARSEC, and GraphBIG benchmarks. To understand runtime behaviors of workloads, we classify the variations on two homogeneous processors that

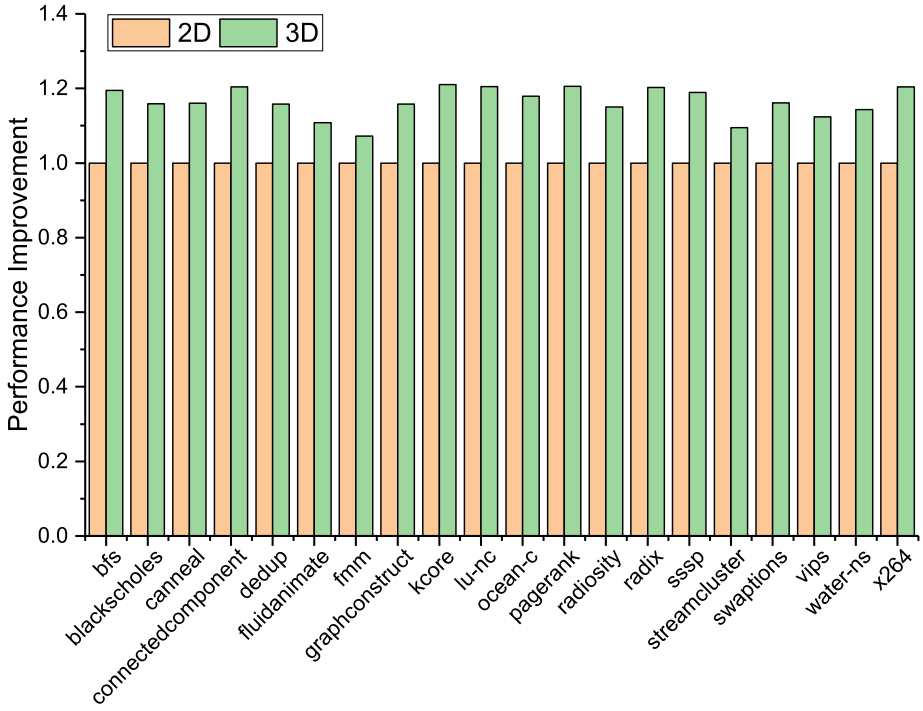


Figure 6.8: Performance comparison of a single in-order core in a planar and 3D design

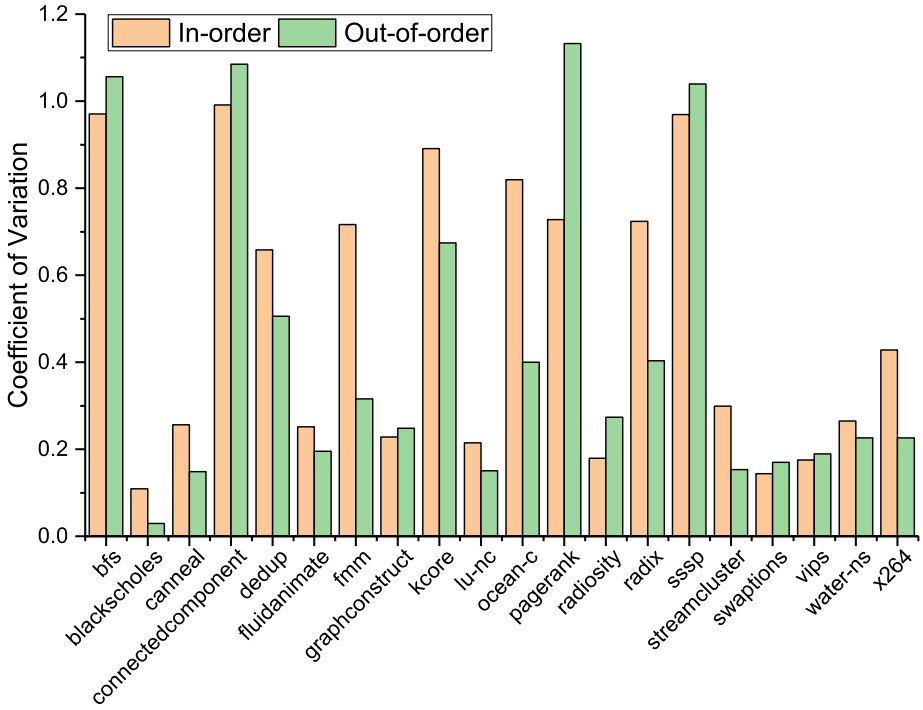


Figure 6.9: Coefficient of variation between an in-order and out-of-order core

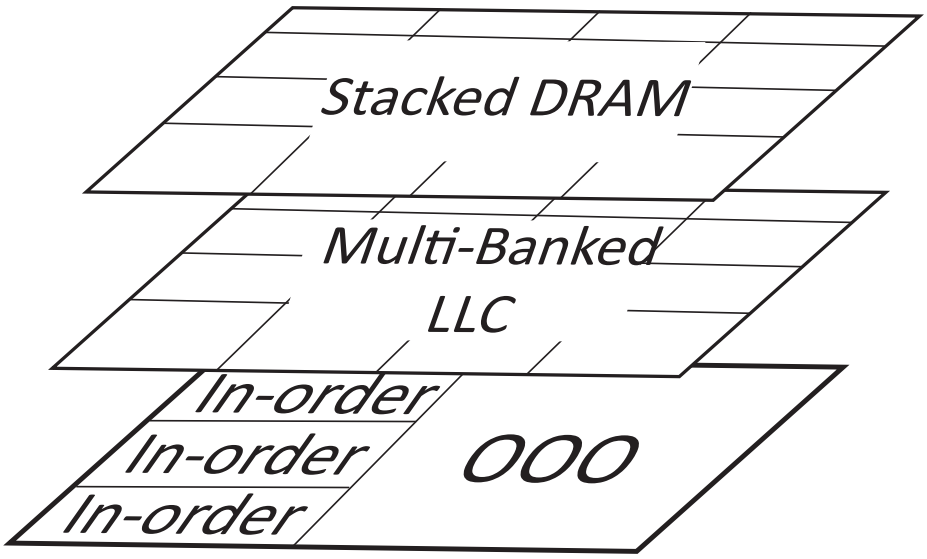


Figure 6.10: Architecture of a 3D asymmetric tiled processor

motivate the use of fine grained schedulers. Specifically, we compare the performance and power consumption between a 4-core in-order and single out-of-order processor, and between an 8-core in-order and 2-core out-of-order processor. Simulation results are given in Figure 6.11 to Figure 6.14.

As shown in Figure 6.11 and Figure 6.13, the performance of a out-of-order core is comparable to that of 4 in-order cores. For high IPC applications with few inter-core communications such as $radiosity$ and $streamcluster$, in-order cores outperform the out-of-order core. For memory bounded applications $lu\text{-}nc$ and $ocean\text{-}c$, out-of-order cores are better, as the out-of-order implementation hides the long latency of memory operations by exploring the instruction-level parallelism.

The power comparison between in-order and out-of-order cores are presented in Figure 6.12 and Figure 6.14. Basically, the total power in out-of-order cores surpasses that of in-order cores, since out-of-order design is optimized for latency and proven not as power efficient. Notice that the in-order core power is higher in applications such as $radiosity$ and $streamcluster$. This is because the performance gain in these applications using larger number of cores is greater than an out-of-order core, and thus consumes more power.

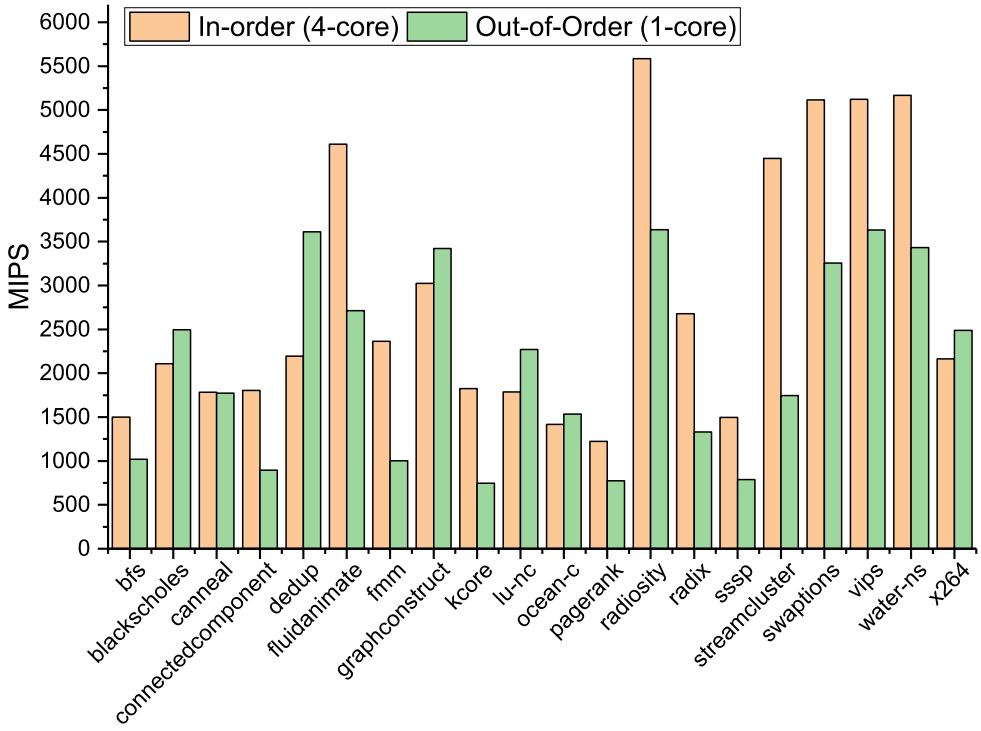


Figure 6.11: Performance comparison between a 4-core in-order and single core out-of-order processor

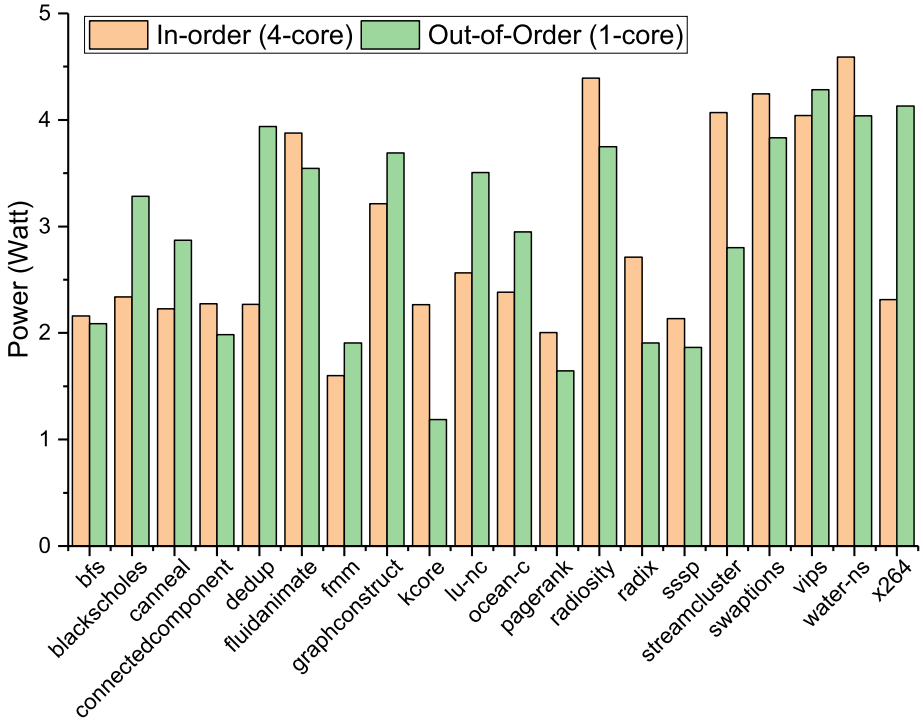


Figure 6.12: Power comparison between a 4-core in-order and single core out-of-order processor

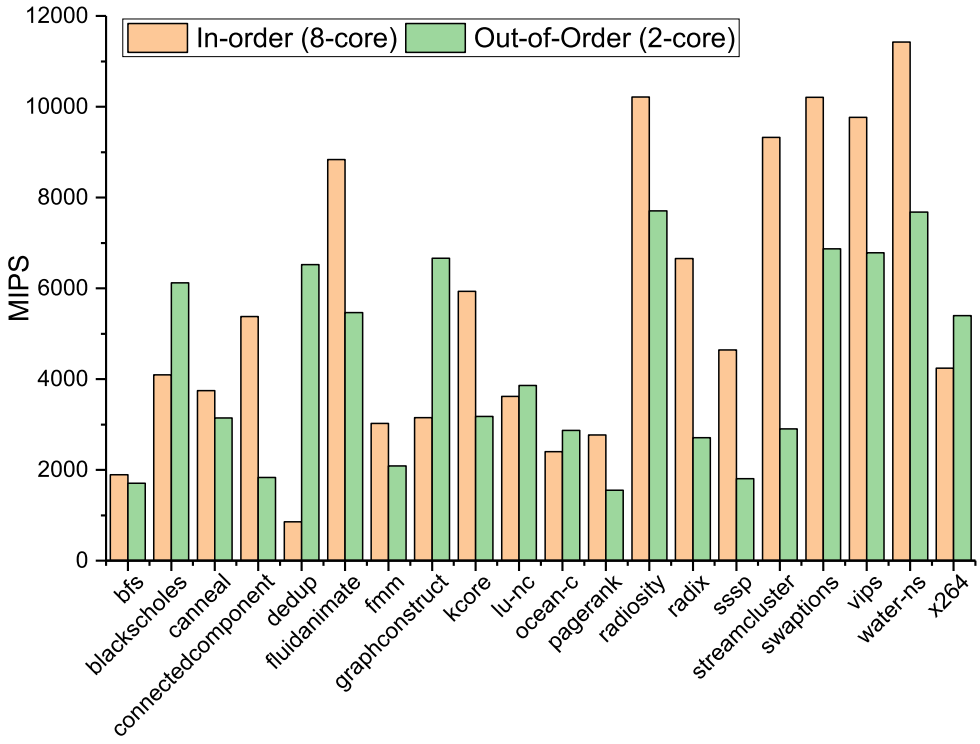


Figure 6.13: Performance comparison between an 8-core in-order and 2-core out-of-order processor

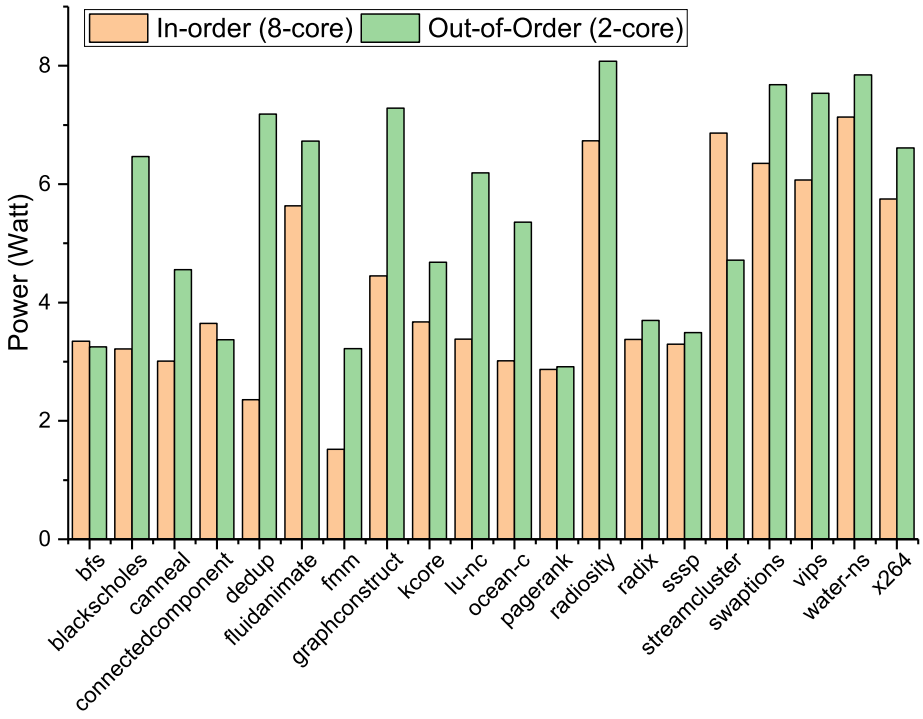


Figure 6.14: Power comparison between an 8-core in-order and 2-core out-of-order processor

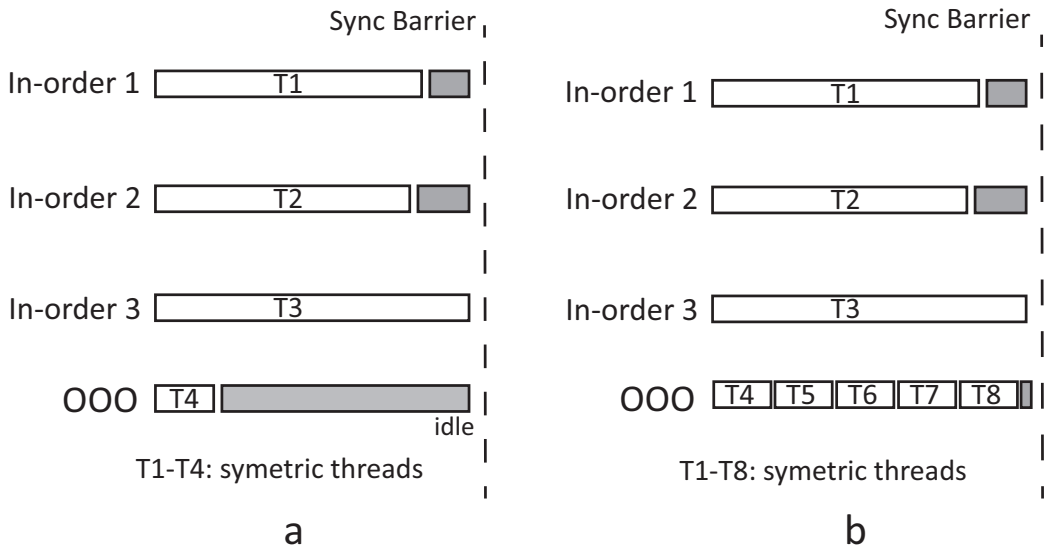


Figure 6.15: Thread scheduling in (a) an unbalanced workload, and (b) balanced workload between in-order and out-of-order cores

6.3.3 Workload Optimization in a Heterogeneous Processor

As described in chapter 3, the Manifold simulator supports multi-threading enabled by Linux symmetric multiprocessing. Therefore, the guest Linux system does not aware of the underlining asymmetric processor design, leading to system inefficiency. Figure 6.15 (a) depicts one scenario of such ineffective scheduling. T1 $\sim$ T4 are four threads spawned from an application with similar workload. T4 is scheduled to the out-of-order core while other threads are scheduled to in-order cores. The out-of-order core has to wait for a quite long period before T3 reaches the synchronization barrier.

To improve system efficiency, we must balance workloads assigned between in-order and out-of-order cores. Instead of modifying the scheduling policy of the guest Linux, we implement a SMP design for the out-of-order core that fetches instructions from multiple threads simultaneously in a cycle, as shown in Figure 6.15 (b). By assigning multiple threads to the out-of-order core, we maximize the utilization in out-of-order cores and reduce overall system idle time. Figure 6.16 compares the idle time in the out-of-order pipeline during execution. In the baseline configuration, threads are evenly distributed

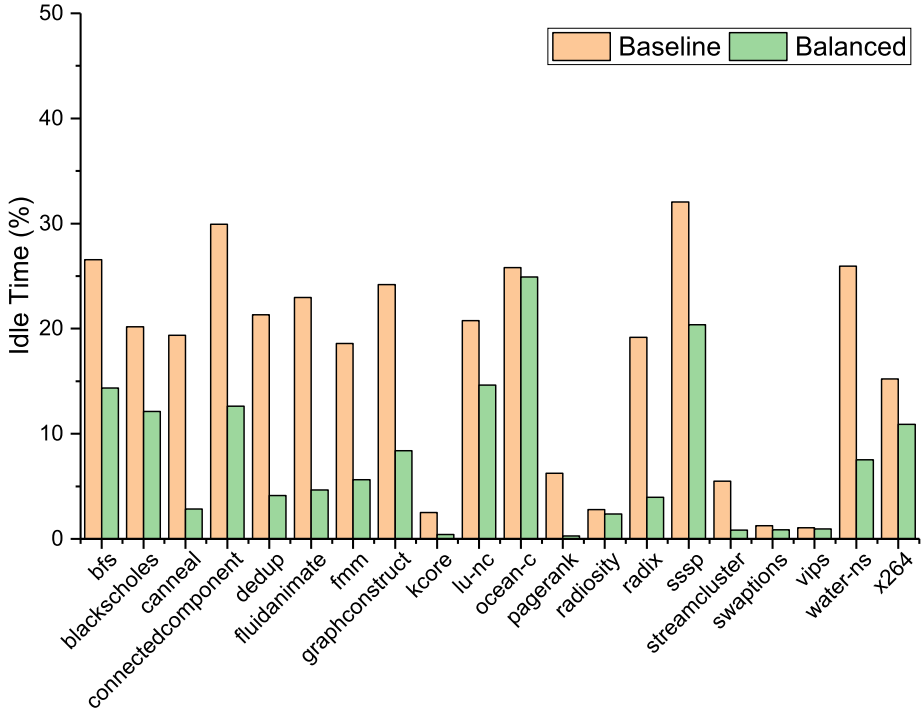


Figure 6.16: Idle time comparison of the out-of-order core in a SMP design

among cores regardless the core type; in the balanced design, the out-of-order cores are assigned multiple (in this example, 5) threads so as to balance in-order cores' workload.

Simulation results indicate that the workload optimization in the balanced design can significantly reduces system idle time. Compared to the unbalanced case, workload optimization saves on average 65% idle time of the out-of-order core. The SMP design in the out-of-order core especially benefits applications with significant discrepancy in performance between in-order and out-of-order cores such as $streamclusters$ and $pagerank$.

6.3.4 Thread Utility Based Scheduling

As the workloads exhibit a wide range of performance and power variation, we are motivated by a runtime scheduling to maximize performance and power efficiency. Thread scheduling for heterogeneous processors has been an active area of research [96] [98] [99] [100]. However, the focus of this work is the design of a power efficient heterogeneous processor with respect to the characteristics of 3D packages, and we use thread scheduling as

the means to explore the effectiveness of 3D heterogeneous design to maximize the system performance within a given thermal cap. Thus, while this thesis does not claim adavances in power efficient scheduling, we construct a general framework for thread scheduling that explores the effectiveness of the combination of in-order and out-of-order cores and that can make use of existing scheduling algorithms.

The scheduling framework tracks the execution time of each thread spending on out-of-order cores defined as $thread\ utility$. A vector U_i is used to represent the time $thread_i$ executes on each out-of-order core. Thus $U_i = [u_{i1}u_{i2}...u_{im}]$, where m is the number of out-of-order cores in the processor.

We formulate an optimization problem for thread scheduling in a 3D heterogeneous processor based on the concept of thread utility. We define a cost function $f(U)$ such that under the power (TDP) and thermal (T_{max}) constraints we can find:

$$\begin{aligned} & min_U f(U) \\ & s.t. \quad power(U) < TDP, \\ & \qquad max(T(U)) < T_{max}, \\ & \sum_{i=1}^{n} u_{ji} = 1, \quad for\, all\, j = 1\, to\, m. \end{aligned} \tag{6.2}$$

$T(\cdot)$ is a temperature function with respect to thread utility. We solve above equations to get an optimal utility matrix $U = [U_1, U_2, ..., U_n]$ that defines the priority of threads scheduled to the out-of-order core. When U_i is below a threshold, thread T_i is migrated to in-order cores for threads with high utility. A snapshot of thread scheduling is shown in Figure 6.17. When the quota of thread T8 exhausts, it is swapped out to in-order 3 in the next scheduling interval.

From previous analysis, memory bounded applications are more power efficient on the out-of-order cores, while computational bounded applications are more efficient on in-order cores. We simplify the scheduling problem with a runtime metric byte per instruction

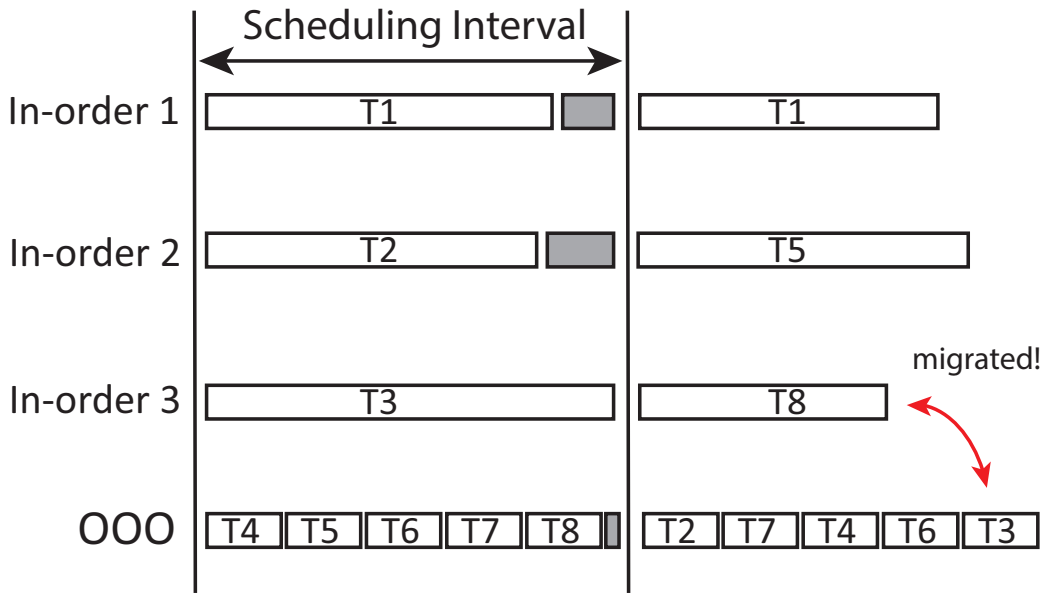


Figure 6.17: Thread utility based scheduling in a 3D heterogeneous processor

(BPI) defined as the ratio of data request (in bytes) sent to the memory system and total instructions in a given sampling interval. We use a fine grained scheduler so that thread behaviors remains roughly the same in consecutive intervals. BPI is used to calculate the quota of each thread for a given scheduling period.

6.3.5 Results and Analysis

We implement a thread utility based scheduler in the 4-core 3D heterogeneous processor called 3DSched and compare it with an 8-core in-order and a 2-core out-of-order homogeneous processor in terms of system performance, power consumption, and energy efficiency.

Figure 6.18 shows the performance comparison of the three systems in terms of MIPS. The performance of 3DSched is comparable to the out-of-core processor for memory bounded applications (e.g., $dedup$) and to the in-order cores for computational bounded applications (e.g., $radiosity$). As 3DSched optimizes the workload balance dynamically, it obtains best performance in certain applications such as $lu\text{-}nc$ and bfs.

Figure 6.19 compares runtime power of the three processors. 3DSched shows a moder-

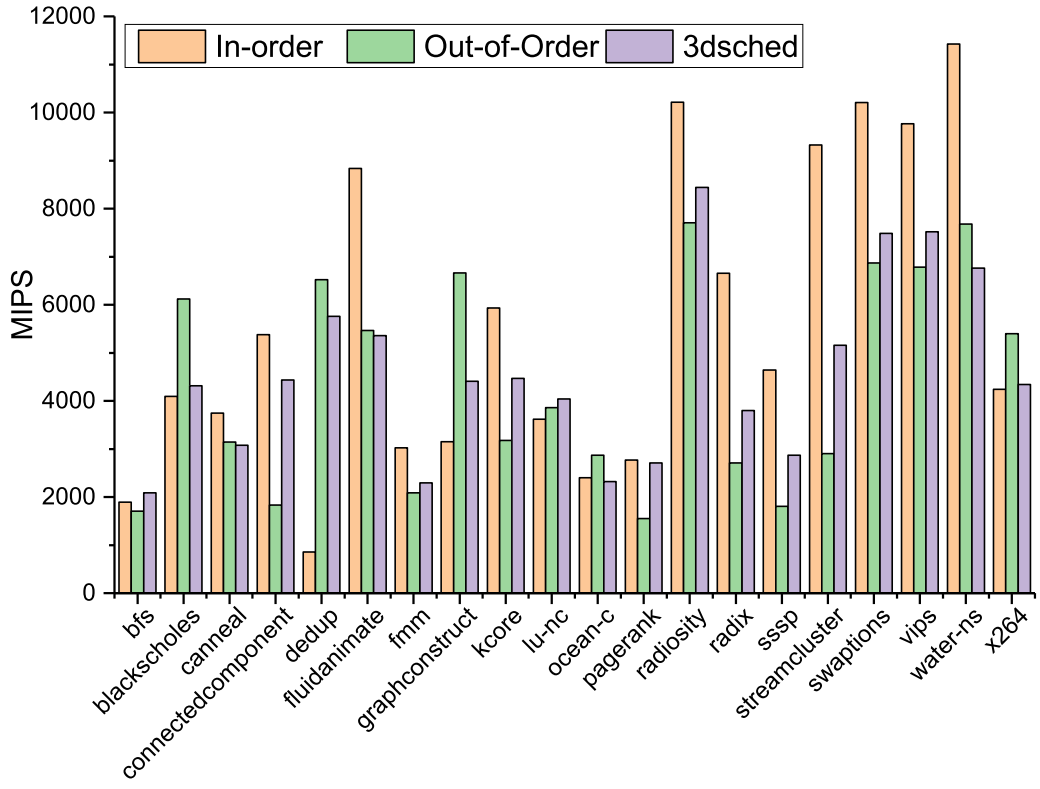


Figure 6.18: Performance comparison using the 3DSched scheduler

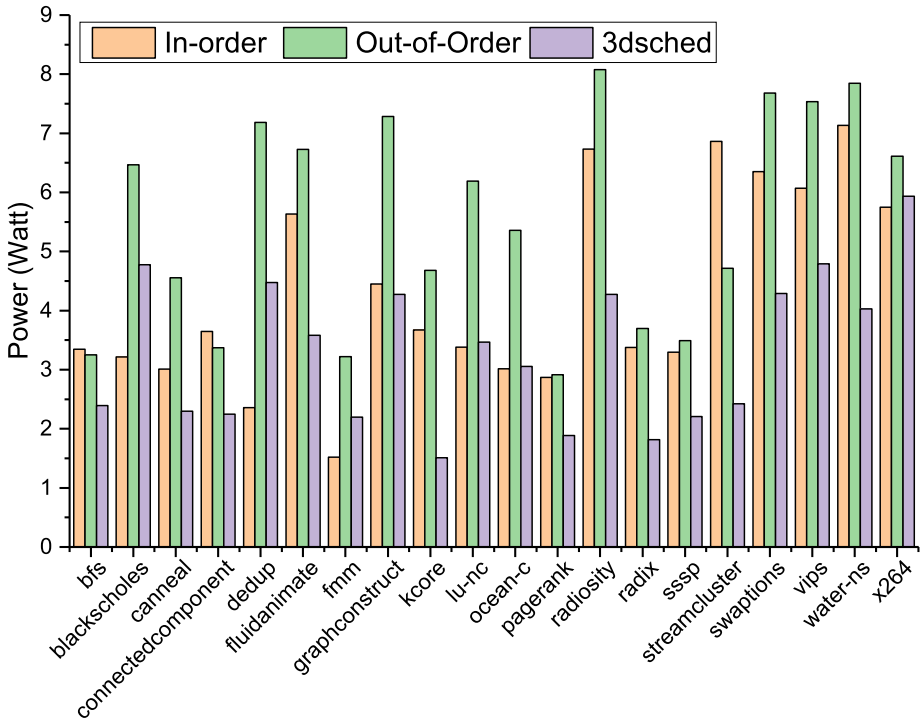


Figure 6.19: Power comparison using the 3DSched scheduler

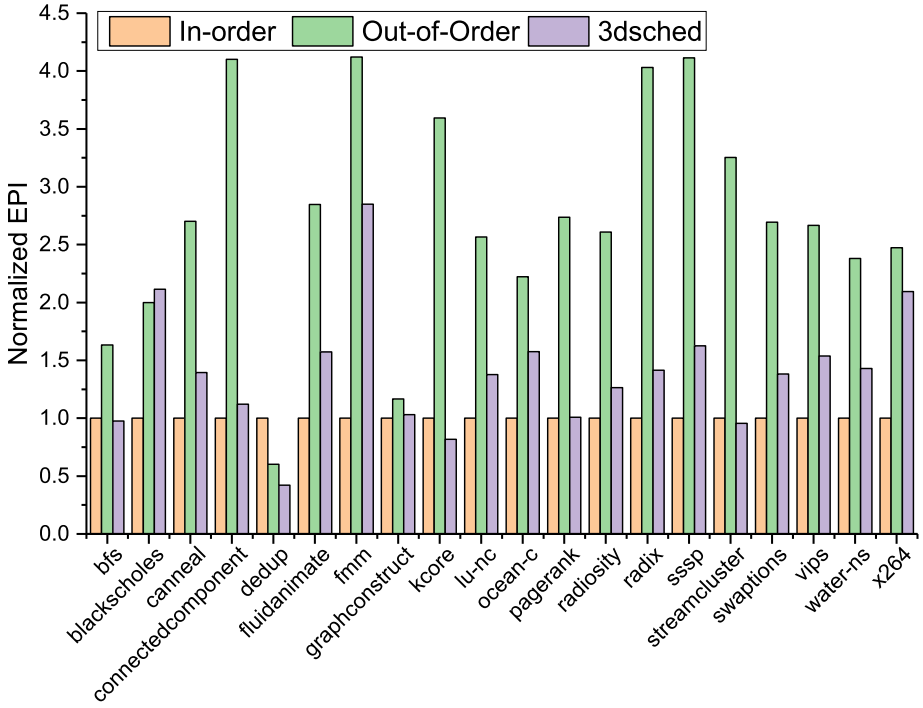


Figure 6.20: Energy efficiency comparison using the 3DSched scheduler

ated power for all applications. The total power consumption of 3DSched is between that of the in-order and out-of-order processors.

Finally, we compare the system energy efficiency in terms of normalized EPI, as demonstrated in Figure 6.20. As the thread utility scheduler optimizes power efficiency dynamically based on the executing information of application threads, EPI of 3DSched is comparable to in-order processors, and system efficiency is much improved compared to an out-of-order processors throughout tested applications.

6.4 Summary

In this chapter, we investigate the refactored memory hierarchy in 3D packages and seek for opportunities in optimizing the processor design with respect to performance, power consumption, and energy efficiency based on the characteristics of 3D packages.

In the first part of this chapter, we look into the pin bandwidth constraints that limit the progress of large-scale processor design and propose a 2-tier Short-Stack structure to

address this problem using FinFETbased eDRAM cells as the system LLC. Short-Stack brings significant performance improvement due to the substantial increase of the LLC bandwidth, and improves over 25% average performance gain compared to the 2D baseline. On the other hand, the larger LLC capacity and cache associativity enabled by the eDRAM implementation reduces the miss rate and thus the demand of off-chip memory bandwidth. Moreover, we model more different system configurations using SRAM and eDRAM LLC in detail and study the impact of Short-Stack. Full system simulation results show that the Short-Stack structure with eDRAM LLC saves 5.6% of energy consumption on average and improves approximately 4% of energy efficiency, suggesting the ss-eDRAM is a viable alternative for future processor designs.

In the second part of the chapter, we motivate the necessity and use of heterogeneous processor design in 3D processors to maximize system power efficiency given the limitations of power supply and cooling capability. As the memory system is improved in 3D packages, replacing one out-of-order core with multiple in-order core alternatives can significantly reduce the power consumption without compromising system performance. Furthermore, we deploy a scheduling algorithm based on thread utility that identifies the patterns of thread execution between in-order and out-of-order cores to further improve the system performance in our proposed heterogeneous 3D processor. Specifically, the algorithm schedules threads based on the runtime information of each thread for the optimization in power efficiency with thermal constraints. We also implement an SMP design in the asymmetric architecture to balance the workload between the in-order and out-of-order cores. Simulation results indicate the effectiveness of applying thread scheduling in such heterogeneous 3D processors to gain performance improvement and power efficiency.

CHAPTER 7
CONCLUSION

This dissertation focuses on high performance and energy efficient multi-core processor design in 3D integrated circuits. As performance, power, and thermal are strongly coupled, 3D multi-/many- core processors introduce great challenges to processor designers. We revisit the lower-level circuit and physical interactions in a 3D processor based on our multi-physics models of microarchitectural components and identify the importance of an adaptive system design through the holistic multi-physics co-optimizations. This is critical to achieve performance and efficiency improvements in 3D processors . In this dissertation, we explore and discuss three aspects of the co-design practice:

1. Co-optimizing microfluidic cooling with processor floorplan and power map and proposing two high-performance thermally-adaptive processor designs;
2. Minimizing the thermal guardband of the supply voltage in a 3D last-level cache and reducing the transient voltage variations in an on-chip voltage regulator with a learning-based predication of voltage emergencies;
3. Addressing the pin stress problem with a Short-Stack structure that reduces off-chip data requests and exploring a power-efficient heterogeneous multi-core processor based on the characterization of 3D packages.

In conclusion, this dissertation proposes several techniques in 3D processor design to explore the co-design opportunities of low-level physics and processor microarchitecture. Compared to the practice of the worst-case design paradigm, these system-level optimizations prove the essence of deploying adaptive processor design to obtain significant improvements in thermal hotspot reduction, system performance gain, and power/energy efficiency in 3D ICs. This dissertation hopes to establish the role of such co-design practices in 3D processors and to contribute as the guidelines for future large-scale processors.

Appendices

APPENDIX A

COMPACT THERMAL MODEL FOR 3D MICROFLUIDIC COOLING

The compact thermal model is constructed from the geometry of the processor shown in Figure A.1. Each tier has three layers: a metal and SiO2 layer for bonding and routing, an active device layer, and a silicon substrate. The circular pin fin is fabricated on the back of the chip by Deep Reactive-Ion Etching and signal TSVs are embedded inside the pin fins for inter-tier data communication. Natural convection is assumed on top of the chip stack.

The geometric model is discretized into interconnected control volumes. From each control volume, the analysis in energy balance is conducted in terms of energy equations of a unit control volume around a single pin. Specifically, we separate the solid and fluid domains and carry out the thermal analysis with a finite-element method.

We assume a uniform temperature for each active layer in one control volume for simplification, and the energy equation for the solid domain is

$$Solid : \dot{q}_{gen} + \dot{q}_{cond} + \dot{q}_{conv} = 0, \tag{A.1}$$

where $\dot{q}_{gen}$ is the energy generation term from the power map, $\dot{q}_{cond}$ the heat conduction from neighboring control volumes, and $\dot{q}_{cond}$ the heat transferred by convection.

For the fluid flow, we assume one direction flow and neglect the axial conduction inside the fluid. In our experiment, we choose DI-water as the coolant due to its good thermal performance for single phase cooling. The energy balance equation for the fluid domain is

$$Fluid : \dot{m} \cdot C_p \cdot (T_{f,in} - T_{f,out}) + \dot{q}_{conv} = 0, \tag{A.2}$$

where $\dot{m}$ is the mass flow, C_p heat capacity, $T_{f,in}$ and $T_{f,out}$ the liquid temperature of the inlet and outlet flow, and $\dot{q}_{cond}$ the heat transferred by convection.

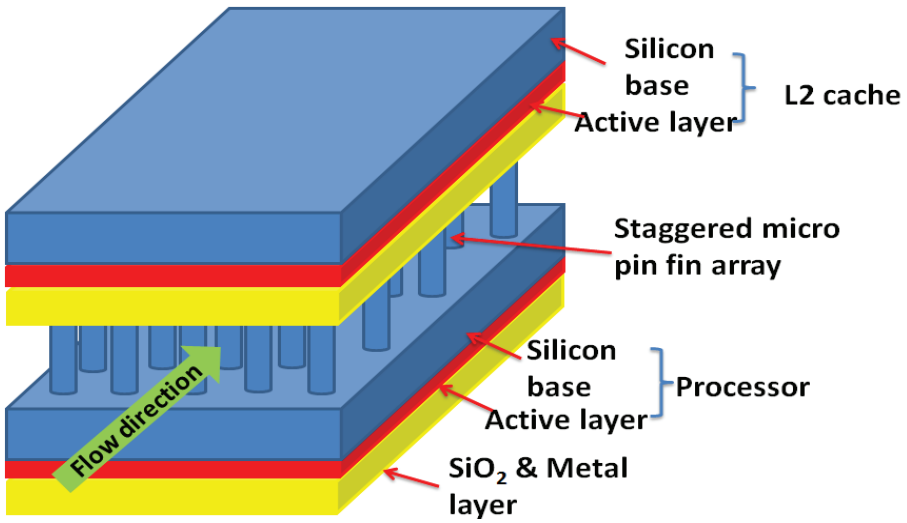


Figure A.1: The geometric model of 3D stacked ICs with microfluidic cooling [3]

Table A.1: Material properties used in the thermal simulation

Materials	SiO2&Metal	Si base	Pin fin
Length(mm)	8.4	8.4	initial*
Width(mm)	8.4	8.4	
Thickness(μm)	10	100	
Thermal conductivity ($W/(mK)$)	1.4	149	149

* Initial pin diameter $100\mu m$, pin height and pitch is $200\mu m$.

Table A.2: Heatsink parameters used in the thermal simulation

Air convection heatsink	
Heat transfer coefficient	$1.2e^{-11}W/\mu m^2K$
Ambient temperature	$300K$
Micro-pin fin heatsink	
Coolant volumetric heat capacity	$4.17e^{-11}J/\mu m^3K$
Pin distribution	staggered

We obtain the temperature distribution by solving the above energy equations simultaneously. Due to the special nature of the energy flow [101], we only discuss the conduction between the core and LLC and between the fluid and solid for simplification in the compact thermal model of 3D processors.

Table A.1 and Table A.2 list material properties and cooling parameters for the thermal analysis used in the 3D compact thermal model.

APPENDIX B

EDRAM HSPICE MODEL IN A 3D PACKAGE

An eDRAM simulation framework is implemented for eDRAM parametric analysis in HSPICE, as demonstrated in Figure B.1 (a), and focus on a high-performance gaincell modeled as a two-transistor (2T) NFET design optimized for cell density. The improvements in cell density comes from the removal of the well keep-out area, as all eDRAM cells are built in the same substrate [92].

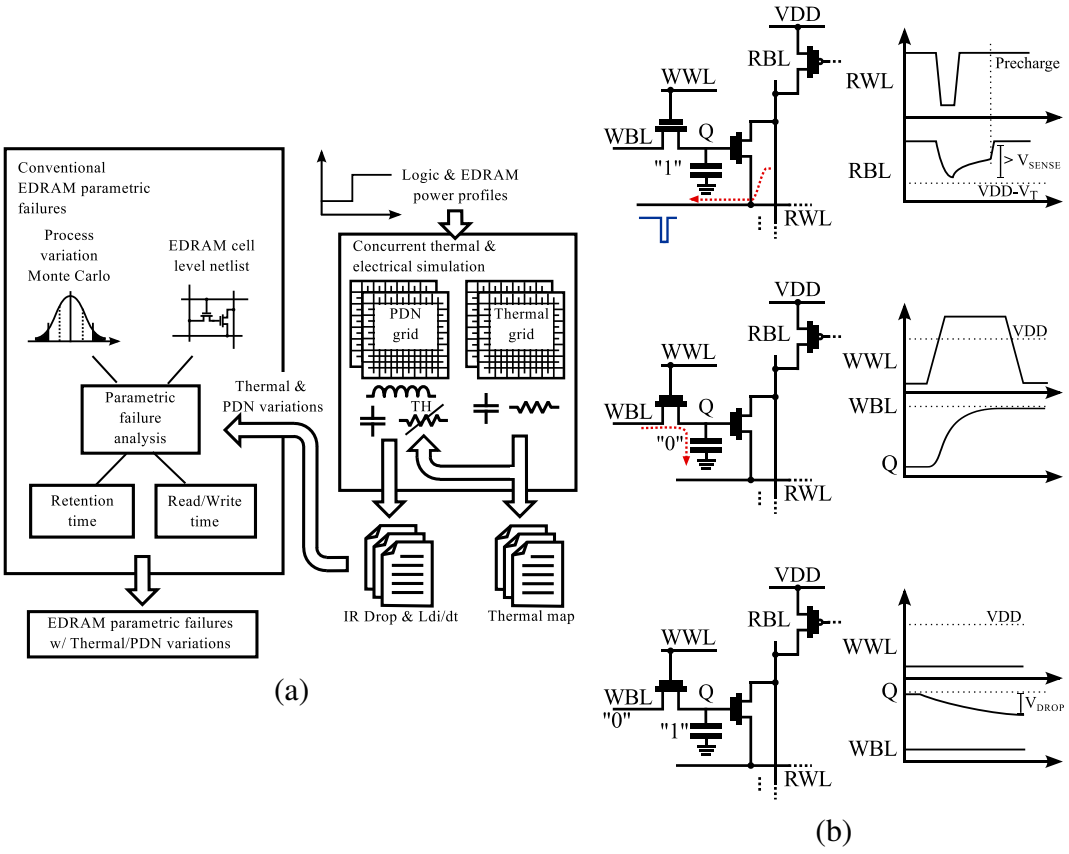


Figure B.1: The simulation methodology for thermal and supply cross-talk aware eDRAM analysis: (a) the co-simulation framework of supply and thermal grids [102], and (b) eDRAM operations [5]

Operations of eDRAM cells are depicted in Figure B.1 (b), indicating a similar critical path between eDRAM and traditional 6T SRAM sub-banks. During a read operation, the

Table B.1: Transistor ratio of a SRAM cell

Transistors	Planar 45 nm (ratio)	FinFET 16 nm (ratio)
PULLUP	1	1
ACCESS	1.5	1
PULLDN	2	2

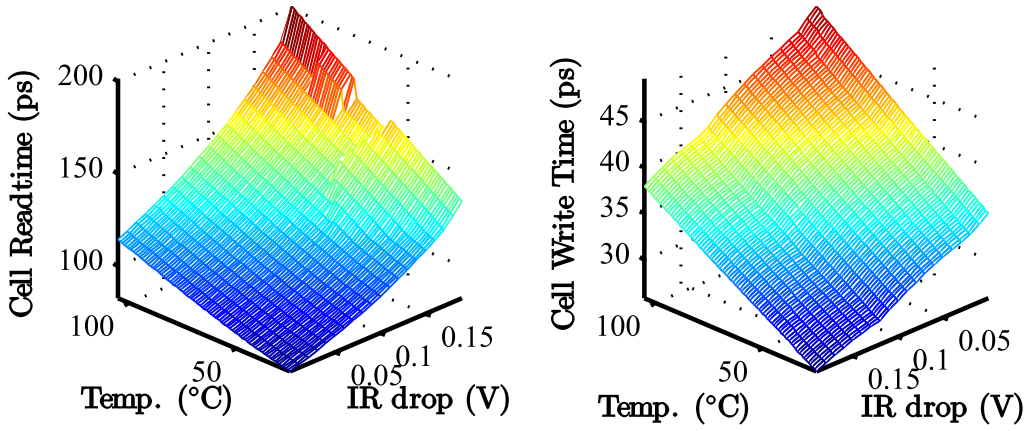
Table B.2: Transistor Ratio of a eDRAM cell

Transistors	Planar 45 nm (ratio)	FinFET 16 nm (ratio)
WRITE	2	2
READ	1	1

flop-to-flop delay of the eDRAM cell is defined by the read wordline driver, cell drive bit-line, sensamp sensing, and read bit-line precharge/sensamp reset. The write operation uses a single NFET and thus a write cycle is much simpler than a read in that its delay is determined only by the wordline and bitline drivers. Due to the non-regenerative charge inside the cell, eDRAM cells are required to refresh constantly. In our model, cell retention time is defined as the time of a fully charged cell discharged to $Vdd/2 + 100mV$.

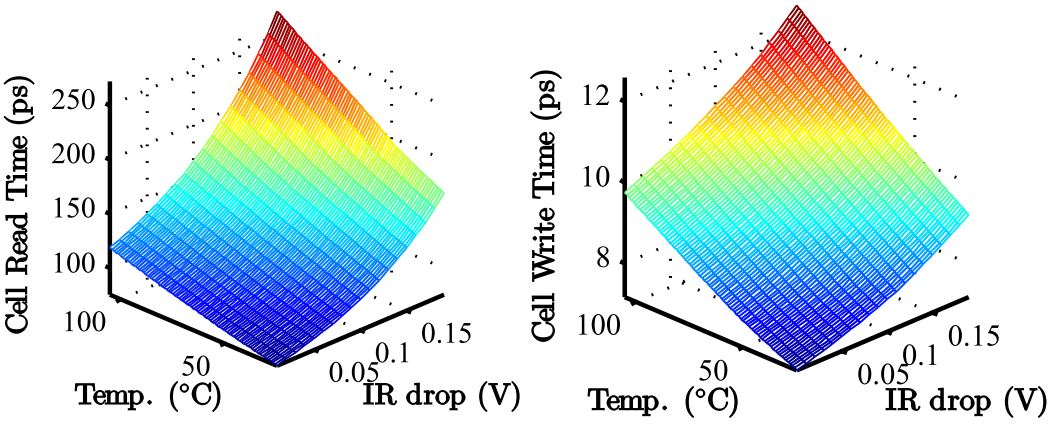
We evaluate the performance of eDRAM with traditional 6T SRAM at cell level. The corresponding sizes are shown in Table B.1 for SRAM and Table B.2 for eDRAM. The sub-array row on the critical path is 32 cells on a single bitline. Figure B.2 and Figure B.3 demonstrate that the read and write time of memory cells in a planar design are more sensitive to supply droop than cell temperature in the given operating region. More interestingly, the delay response negatively correlate with temperature in FinFET devices, as shown in Figure B.4 and Figure B.5. The planar device operates slower at higher temperature but the FinFET operates slightly faster due to V_{T}'s temperature dependency. This technology dependent correlation may be applied to allow for additional thermal optimization.

Comparing with SRAM operations in Figure B.3 and Figure B.5, the eDRAM read time is comparable to that in SRAM with the same array configuration with only 8% read delay penalty. Meanwhile, eDRAM is 60% slower comparing to the SRAM in planar cell design. Since the read operation defines the flop-to-flop critical path, the inferior eDRAM versus SRAM write delay is masked by the longer read access in a random access cycle.



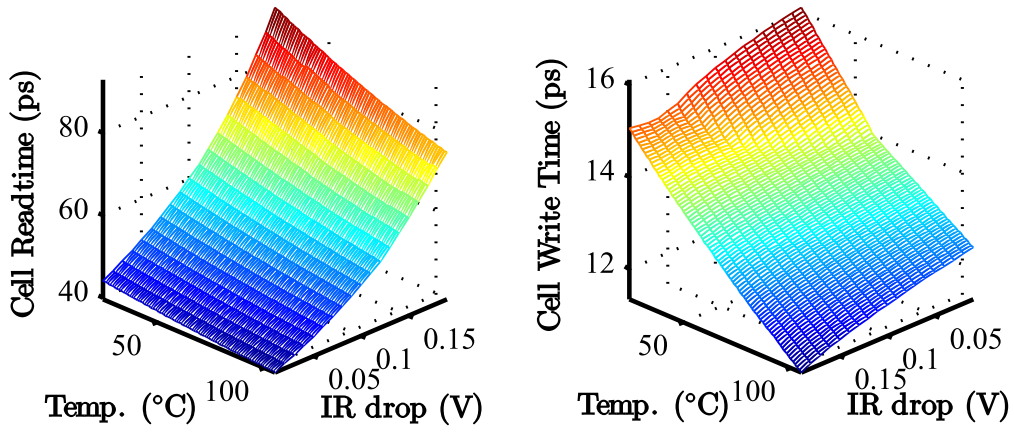
(a) (b)

Figure B.2: Temperature sensitive delay in planar eDRAM (a) read time, (b) write time [102]



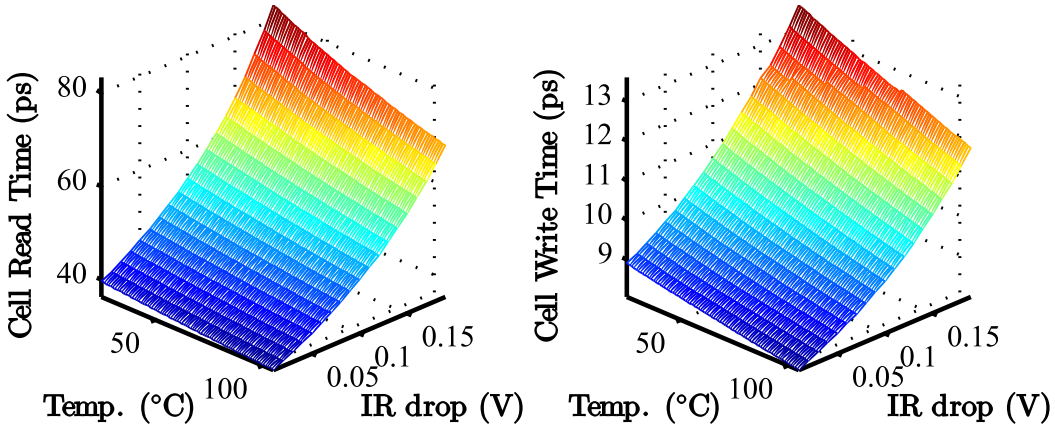
(a) (b)

Figure B.3: Temperature sensitive delay in planar SRAM: (a) read time, (b) write time [102]



(a) (b)

Figure B.4: Temperature sensitive delay in FinFET eDRAM: (a) read time, (b) write time [5]



(a) (b)

Figure B.5: Temperature sensitive delay in FinFET SRAM: (a) read time, (b) write time [5]

REFERENCES

[1] J. Wang, J. Beu, R. Bheda, T. Conte, Z. Dong, C. Kersey, M. Rasquinha, G. Riley, W. Song, H. Xiao, P. Xu, and S. Yalamanchili, "Manifold: A parallel simulation framework for multicore systems," in *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2014, pp. 106–115.

[2] H. Xiao, W. Yueh, S. Mukhopadhyay, and S. Yalamanchili, "Thermally adaptive cache access mechanisms for 3d many-core architectures," *IEEE Computer Architecture Letters*, vol. 15, no. 2, pp. 129–132, 2016.

[3] H. Xiao, Z. Wan, S. Yalamanchili, and Y. Joshi, "Leakage power characterization and minimization in 3d stacked multi-core chips with microfluidic cooling," in *2014 Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, 2014, pp. 207–212.

[4] H. Xiao, W. Yueh, S. Mukhopadhyay, and S. Yalamanchili, "Multi-physics driven co-design of 3d multicore architectures," *ASME. International Electronic Packaging Technical Conference and Exhibition*, vol. 1: Thermal Management, 2015.

[5] H. Xiao, W. Yueh, S. Mukhopadhyay, and S. Yalamanchili, "Short-stack: Pushing back the pin bandwidth wall with finfet-based edram in-package last-level cache," in *2015 Semiconductor Research Corporation TECHCON conference*, 2015.

[6] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ser. ISCA '11, San Jose, California, USA, 2011, pp. 365–376.

[7] Y. Zhang, "Hybrid microfluidic cooling and thermal isolation technologies for 3d ics," PhD thesis, Georgia Institute of Technology, 2015.

[8] N. H. Khan, S. M. Alam, and S. Hassoun, "System-level comparison of power delivery design for 2d and 3d ics," in *2009 IEEE International Conference on 3D System Integration*, 2009, pp. 1–7.

[9] M. Bamal, S. List, M. Stucchi, A. S. Verhulst, M. V. Hove, R. Cartuyvels, G. Beyer, and K. Maex, "Performance comparison of interconnect technology and architecture options for deep submicron technology nodes," in *2006 International Interconnect Technology Conference*, 2006, pp. 202–204.

[10] H. C. Chien, J. H. Lau, Y. L. Chao, M. J. Dai, R. M. Tain, L. Li, P. Su, J. Xue, and M. Brillhart, "Thermal evaluation and analyses of 3d ic integration sip with tsvs for network system applications," in *2012 IEEE 62nd Electronic Components and Technology Conference*, 2012, pp. 1866–1873.

[11] S. M. Sri-Jayantha, G. McVicker, K. Bernstein, and J. U. Knickerbocker, "Thermomechanical modeling of 3d electronic packages," *IBM Journal of Research and Development*, vol. 52, no. 6, pp. 623–634, 2008.

[12] Y. Zhang, C. R. King, J. Zaveri, Y. J. Kim, V. Sahu, Y. Joshi, and M. S. Bakir, "Coupled electrical and thermal 3d ic centric microfluidic heat sink design and technology," in *2011 IEEE 61st Electronic Components and Technology Conference (ECTC)*, 2011, pp. 2037–2044.

[13] B. A. Jasperson, Y. Jeon, K. T. Turner, F. E. Pfefferkorn, and W. Qu, "Comparison of micro-pin-fin and microchannel heat sinks considering thermal-hydraulic performance and manufacturability," *IEEE Transactions on Components and Packaging Technologies*, vol. 33, no. 1, pp. 148–160, 2010.

[14] S. Ndao, Y. Peles, and M. K. Jensen, "Multi-objective thermal design optimization and comparative analysis of electronics cooling technologies," *International Journal of Heat and Mass Transfer*, vol. 52, no. 1920, pp. 4317 –4326, 2009.

[15] "Optimal arrays of pin fins and plate fins in laminar forced convection," *ASME. Journal of Heat Transfer*, vol. 115, no. 1, pp. 75 –81, 1999.

[16] Z. Wan, H. Xiao, Y. Joshi, and S. Yalamanchili, "Co-design of multicore architectures and microfluidic cooling for 3d stacked ics," *Microelectron. J.*, vol. 45, no. 12, pp. 1814–1821, Dec. 2014.

[17] T. E. Sarvey, Y. Zhang, Y. Zhang, H. Oh, and M. S. Bakir, "Thermal and electrical effects of staggered micropin-fin dimensions for cooling of 3d microsystems," in *Fourteenth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, 2014, pp. 205–212.

[18] K. Chen, I. Hsu, and C. Lee, "Chip-package-pcb thermal co-design for hot spot analysis in soc," in *2012 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, 2012, pp. 215–218.

[19] J. K. John, J. S. Hu, and S. G. Ziavras, "Optimizing the thermal behavior of subarrayed data caches," in *2005 International Conference on Computer Design*, 2005, pp. 625–630.

[20] Z. Jia, Y. Li, Y. Wang, M. Wang, and Z. Shao, "Temperature-aware data allocation for embedded systems with cache and scratchpad memory," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 2, 30:1–30:24, Mar. 2015.

[21] Z. Wang, "Thermal-aware task scheduling on multicore processors," AAI3569706, PhD thesis, Gainesville, FL, USA, 2012, ISBN: 978-1-303-07097-6.

[22] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996, ISBN: 0-13-178609-1.

[23] A. Pirbadian, M. S. Khairy, A. M. Eltawil, and F. J. Kurdahi, "State dependent statistical timing model for voltage scaled circuits," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 1432–1435.

[24] Q. Zhu, *Power Distribution Network Design for VLSI*. Wiley, 2004.

[25] Y. J. Lee and S. K. Lim, "Timing analysis and optimization for 3d stacked multi-core microprocessors," in *2010 IEEE International 3D Systems Integration Conference (3DIC)*, 2010, pp. 1–7.

[26] H. He, "Quantitative Analysis and Modeling of 3-D TSV-Based Power Delivery Architectures," PhD thesis, Rensselaer Polytechnic Institute, 2015.

[27] H. J. Zhang. (2013). Basic concepts of linear regulator and switching mode power supplies. Accessed: 2017-02-10.

[28] K. Onizuka, K. Inagaki, H. Kawaguchi, M. Takamiya, and T. Sakurai, "Stacked-chip implementation of on-chip buck converter for distributed power supply system in sips," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 11, pp. 2404–2410, 2007.

[29] J. Sun, D. Giuliano, S. Devarajan, J. Q. Lu, T. P. Chow, and R. J. Gutmann, "Fully monolithic cellular buck converter design for 3-d power delivery," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 3, pp. 447–451, 2009.

[30] S. Carlo, W. Yueh, and S. Mukhopadhyay, "On the potential of 3d integration of inductive dc-dc converter for high-performance power delivery," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013, pp. 1–8.

[31] Z. Li, Y. Ma, Q. Zhou, Y. Cai, Y. Wang, T. Huang, and Y. Xie, "Thermal-aware power network design for ir drop reduction in 3d ics," in *17th Asia and South Pacific Design Automation Conference*, 2012, pp. 47–52.

[32] T. Song and S. K. Lim, "A fine-grained co-simulation methodology for ir-drop noise in silicon interposer and tsv-based 3d ic," in *2011 IEEE 20th Conference on Electrical Performance of Electronic Packaging and Systems*, 2011, pp. 239–242.

[33] R. Berthiaume, "Voltage limit: Processor lives depend on it," *Electronics Systems and Software*, vol. 2, no. 2, pp. 28–32, 2004.

[34] E. Vosicher and E. Lougee, "Hysteretic controller fits processor needs," *PCIM Power Electronic Systems*, vol. 26, no. 1, pp. 28 –40, 2000/01/.

[35] X. Hu, Y. Xu, Y. Hu, and Y. Xie, "Swimminglane: A composite approach to mitigate voltage droop effects in 3d power delivery network," in *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014, pp. 550–555.

[36] V. J. Reddi, M. S. Gupta, G. Holloway, G. Y. Wei, M. D. Smith, and D. Brooks, "Voltage emergency prediction: Using signatures to reduce operating margins," in *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, 2009, pp. 18–29.

[37] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, and V. J. Reddi, "Safe limits on voltage reduction efficiency in gpus: A direct measurement approach," in *Proceedings of the 48th International Symposium on Microarchitecture*, ser. MICRO-48, Waikiki, Hawaii: ACM, 2015, pp. 294–307, ISBN: 978-1-4503-4034-2.

[38] Y. Kim, L. K. John, I. Paul, S. Manne, and M. Schulte, "Performance boosting under reliability and power constraints," in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 334–341.

[39] W. J. Yeon, "Dynamic voltage and frequency scaling techniques for chip multiprocessor designs," pp. 105 –, 2015//.

[40] C. Torng, M. Wang, and C. Batten, "Asymmetry-aware work-stealing runtimes," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 40–52.

[41] P. Rathnala, A. Kharaz, and T. Wilmshurst, "An efficient adaptive voltage scaling using delay monitor unit," in *2015 11th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, 2015, pp. 109–112.

[42] P. Stanley-Marbell, V. C. Cabezas, and R. P. Luijten, "Pinned to the walls: Impact of packaging and application properties on the memory and power walls," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, 2011, pp. 51–56.

[43] J. Macri, "Amd's next generation gpu and high bandwidth memory architecture: Fury," in *2015 IEEE Hot Chips 27 Symposium (HCS)*, 2015, pp. 1–26.

[44] R. Mahajan, R. Sankman, N. Patel, D. W. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik, "Embedded multi-die interconnect bridge (emib) – a high density, high bandwidth packaging interconnect," in *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, 2016, pp. 557–565.

[45] D. Zhang, N. Jayasena, A. Lyashevsky, J. L. Greathouse, L. Xu, and M. Ignatowski, "Top-pim: Throughput-oriented programmable processing in memory," in *Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing*, ser. HPDC '14, Vancouver, BC, Canada: ACM, 2014, pp. 85–98, ISBN: 978-1-4503-2749-7.

[46] S. Han, H. Seo, B. Kim, and E. Y. Chung, "Pim architecture exploration for hmc," in *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2016, pp. 635–636.

[47] J. S. Kwon, T. h. Hwang, and D. S. Kim, "Emulation of processing in memory architecture for application development," in *2016 International SoC Design Conference (ISOCC)*, 2016, pp. 183–184.

[48] D. Kim, J. Kung, S. Chai, S. Yalamanchili, and S. Mukhopadhyay, "Neurocube: A programmable digital neuromorphic architecture with high-density 3d memory," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 380–392.

[49] E. Rotenberg, B. H. Dwiel, E. Forbes, Z. Zhang, R. Widialaksono, R. B. R. Chowdhury, N. Tshibangu, S. Lipa, W. R. Davis, and P. D. Franzon, "Rationale for a 3d heterogeneous multi-core processor," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*, 2013, pp. 154–168.

[50] N. Miura, Y. Koizumi, E. Sasaki, Y. Take, H. Matsutani, T. Kuroda, H. Amano, R. Sakamoto, M. Namiki, K. Usami, M. Kondo, and H. Nakamura, "A scalable 3d heterogeneous multi-core processor with inductive-coupling thruchip interface," in *2013 IEEE COOL Chips XVI*, 2013, pp. 1–3.

[51] C. D. Kersey, A. Rodrigues, and S. Yalamanchili, "A universal parallel front-end for execution driven microarchitecture simulation," in *Proceedings of the 2012 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, ser. RAPIDO '12, Paris, France: ACM, 2012, pp. 25–32, ISBN: 978-1-4503-1114-4.

[52] W. J. Song, S. Mukhopadhyay, and S. Yalamanchili, "Kitfox: Multiphysics libraries for integrated power, thermal, and reliability simulations of multicore microarchi-

tecture," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 5, no. 11, pp. 1590–1601, 2015.

[53] The Manifold Webiste, `www . manifold . gatech . edu`, [Online; accessed Januray-1-2018].

[54] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," in *Proceedings of the 22Nd Annual International Symposium on Computer Architecture*, ser. ISCA '95, S. Margherita Ligure, Italy: ACM, 1995, pp. 24–36, ISBN: 0-89791-698-0.

[55] C. Bienia, S. Kumar, and K. Li, "Parsec vs. splash-2: A quantitative comparison of two multithreaded benchmark suites on chip-multiprocessors," in *2008 IEEE International Symposium on Workload Characterization*, 2008, pp. 47–56.

[56] L. Nai, Y. Xia, I. G. Tanase, H. Kim, and C.-Y. Lin, "Graphbig: Understanding graph computing in the context of industrial solutions," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15, Austin, Texas: ACM, 2015, 69:1–69:12, ISBN: 978-1-4503-3723-6.

[57] R. C. Chu, R. E. Simons, M. J. Ellsworth, R. R. Schmidt, and V. Cozzolino, "Review of cooling technologies for computer products," *IEEE Transactions on Device and Materials Reliability*, vol. 4, no. 4, pp. 568–585, 2004.

[58] P. Teertstra, M. M. Yovanovich, J. R. Culham, and T. Lemczyk, "Analytical forced convection modeling of plate fin heat sinks," in *Fifteenth Annual IEEE Semiconductor Thermal Measurement and Management Symposium (Cat. No.99CH36306)*, 1999, pp. 34–41.

[59] N. Khan, L. H. Yu, T. S. Pin, S. W. Ho, V. Kripesh, D. Pinjala, J. H. Lau, and T. K. Chuan, "3-d packaging with through-silicon via (tsv) for electrical and fluidic interconnections," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 3, no. 2, pp. 221–228, 2013.

[60] P. Zajac, M. Janicki, M. Szermer, C. Maj, P. Pietrzak, and A. Napieralski, "Cache leakage power estimation using architectural model for 32 nm and 16 nm technology nodes," in *2012 28th Annual IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, 2012, pp. 308–312.

[61] Z. Wan, H. Xiao, Y. Joshi, and S. Yalamanchili, "Co-design of multicore architectures and microfluidic cooling for 3d stacked ics," in *19th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, 2013, pp. 237–242.

[62] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2009, pp. 469–480.

[63] The International Technology Roadmap for Semiconductors (ITRS), 2007, `http://www.itrs.net`, [Online; accessed December-17-2017].

[64] S. J. E. Wilton and N. P. Jouppi, "Cacti: An enhanced cache access and cycle time model," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, 1996.

[65] J. Stein and M. M. Hydeman, "Development and testing of the characteristic curve fan model.," *ASHRAE Transactions*, vol. 110, no. 1, pp. 347 –356, 2004.

[66] K. DeVogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "Modeling the temperature bias of power consumption for nanometer-scale cpus in application processors," in *2014 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV)*, 2014, pp. 172–180.

[67] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao, "Exploring sub-20nm finfet design with predictive technology models," in *DAC Design Automation Conference 2012*, 2012, pp. 283–288.

[68] B. Wicht, T. Nirschl, and D. Schmitt-Landsiedel, "Yield and speed optimization of a latch-type voltage sense amplifier," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1148–1158, 2004.

[69] High Frequency Transistor Primer Thermal Properties (Part III), `www.hp.woodshot.com/hprfhelp/4_downld/lit/other/primer3.pdf`, [Online; accessed December-8-2017].

[70] IBM's Power7+ Processor, `http://semiengineering.com/ibms-power7-processor`, [Online; accessed December-19-2017].

[71] G. Semeraro, D. Albonesi, S. Dropsho, G. Magklis, S. Dwarkadas, and M. Scott, "Dynamic frequency and voltage control for a multiple clock domain microarchitecture," in *Microarchitecture, 2002. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, 2002, pp. 356–367.

[72] A. Raghavan, Y. Luo, A. Chandawalla, M. Papaefthymiou, K. P. Pipe, T. Wenisch, and M. Martin, "Computational sprinting," in *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, 2012, pp. 1–12.

[73] J. Jeddeloh and B. Keeth, "Hybrid memory cube new dram architecture increases density and performance," in *2012 Symposium on VLSI Technology (VLSIT)*, 2012, pp. 87–88.

[74] G. Loh, "3d-stacked memory architectures for multi-core processors," in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, 2008, pp. 453–464.

[75] R. Bertran, A. Buyuktosunoglu, P. Bose, T. J. Slegel, G. Salem, S. Carey, R. F. Rizzolo, and T. Strach, "Voltage noise in multi-core processors: Empirical characterization and optimization opportunities," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 2014, pp. 368–380.

[76] K. P. Sozaski, "Modification of reactive power control algorithm," in *Signal Processing Algorithms, Architectures, Arrangements, and Applications SPA 2010*, 2010, pp. 34–39.

[77] Y. Kuai and P. L. Chapman, "Comprehensive and practical optimization of voltage regulator modules," in *2011 IEEE Power and Energy Conference at Illinois*, 2011, pp. 1–6.

[78] P. J. Liu, Y. M. Lai, P. C. Lee, and H. S. Chen, "Fast-transient dc 8211;dc converter with hysteresis prediction voltage control," *IET Power Electronics*, vol. 10, no. 3, pp. 271–278, 2017.

[79] M. Ueno, K. Ito, Y. Ishizuka, and H. Matsuo, "A current compensation circuit for vrm in the sudden change of the load," in *2005 European Conference on Power Electronics and Applications*, 2005, 9 pp.–P.9.

[80] E. A. Burton, G. Schrom, F. Paillet, J. Douglas, W. J. Lambert, K. Radhakrishnan, and M. J. Hill, "Fivr x2014; fully integrated voltage regulators on 4th generation intel x00ae; core x2122; socs," in *2014 IEEE Applied Power Electronics Conference and Exposition - APEC 2014*, 2014, pp. 432–439.

[81] M. Kar, A. Singh, A. Rajan, V. De, and S. Mukhopadhyay, "An all-digital fully integrated inductive buck regulator with a 250-mhz multi-sampled compensator and a lightweight auto-tuner in 130-nm cmos," *IEEE Journal of Solid-State Circuits*, 2017.

[82] B. Li, L. Peng, and B. Ramadass, "Accurate and efficient processor performance prediction via regression tree based modeling," *J. Syst. Archit.*, vol. 55, no. 10-12, pp. 457–467, Oct. 2009.

[83] A. Kerr, E. Anger, G. Hendry, and S. Yalamanchili, "Eiger: A framework for the automated synthesis of statistical performance models," in *2012 19th International Conference on High Performance Computing*, 2012, pp. 1–6.

[84] C. Zhang, A. Ravindran, K. Datta, A. Mukherjee, and B. Joshi, "A machine learning approach to modeling power and performance of chip multiprocessors," in *2011 IEEE 29th International Conference on Computer Design (ICCD)*, 2011, pp. 45–50.

[85] V. J. Reddi, M. Gupta, G. Holloway, M. D. Smith, G. Y. Wei, and D. Brooks, "Predicting voltage droops using recurring program and microarchitectural event activity," *IEEE Micro*, vol. 30, no. 1, pp. 110–110, 2010.

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[87] E. G. M. de Lacerda, A. C.P.L. F. de Carvalho, and T. B. Ludermir, "A study of cross-validation and bootstrap as objective functions for genetic algorithms," in *VII Brazilian Symposium on Neural Networks, 2002. SBRN 2002. Proceedings.*, 2002, pp. 118–123.

[88] J. R. Struharik, "Implementing decision trees in hardware," in *2011 IEEE 9th International Symposium on Intelligent Systems and Informatics*, 2011, pp. 41–46.

[89] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, 2003, pp. 7–18.

[90] M. Kar, A. Singh, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "Exploiting fully integrated inductive voltage regulators to improve side channel resistance of encryption engines.," in *ISLPED*, 2016, pp. 130–135.

[91] M. T. Chang, P. Rosenfeld, S. L. Lu, and B. Jacob, "Technology comparison for large last-level caches (l3cs): Low-leakage sram, low write-energy stt-ram, and refresh-optimized edram," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013, pp. 143–154.

[92] A. Teman, P. Meinerzhagen, A. Burg, and A. Fish, "Review and classification of gain cell edram implementations," in *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, 2012, pp. 1–5.

[93] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *Electron Devices, IEEE Transactions on*, vol. 53, no. 11, pp. 2816–2823, 2006.

[94] M. Annavaram, E. Grochowski, and J. Shen, "Mitigating amdahl's law through epi throttling," in *Proceedings of the 32Nd Annual International Symposium on Computer Architecture*, ser. ISCA '05, Washington, DC, USA: IEEE Computer Society, 2005, pp. 298–309, ISBN: 0-7695-2270-X.

[95] S. Balakrishnan, R. Rajwar, M. Upton, and K. Lai, "The impact of performance asymmetry in emerging multicore architectures," in *Proceedings of the 32Nd Annual International Symposium on Computer Architecture*, ser. ISCA '05, Washington, DC, USA: IEEE Computer Society, 2005, pp. 506–517, ISBN: 0-7695-2270-X.

[96] R. Kumar, K. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, "Processor power reduction via single-isa heterogeneous multi-core architectures," *IEEE Computer Architecture Letters*, vol. 2, no. 1, pp. 2–2, 2003.

[97] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, P. Dubey, S. Junkins, A. Lake, R. Cavin, R. Espasa, E. Grochowski, T. Juan, M. Abrash, J. Sugerman, and P. Hanrahan, "Larrabee: A many-core x86 architecture for visual computing," *IEEE Micro*, vol. 29, no. 1, pp. 10–21, 2009.

[98] K. Van Craeynest, A. Jaleel, L. Eeckhout, P. Narvaez, and J. Emer, "Scheduling heterogeneous multi-cores through performance impact estimation (pie)," in *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ser. ISCA '12, Portland, Oregon: IEEE Computer Society, 2012, pp. 213–224, ISBN: 978-1-4503-1642-2.

[99] J. Chen and L. K. John, "Efficient program scheduling for heterogeneous multi-core processors," in *Proceedings of the 46th Annual Design Automation Conference*, ser. DAC '09, San Francisco, California: ACM, 2009, pp. 927–930, ISBN: 978-1-60558-497-3.

[100] M. Becchi and P. Crowley, "Dynamic thread assignment on heterogeneous multiprocessor architectures," in *Proceedings of the 3rd Conference on Computing Frontiers*, ser. CF '06, Ischia, Italy: ACM, 2006, pp. 29–40, ISBN: 1-59593-302-6.

[101] F. P. Incropera, *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 2006, ISBN: 0470088400.

[102] W. Yueh, "Thermal-aware task scheduling on multicore processors," PhD thesis, Atlanta, GA, USA, 2015.